

Universidad Complutense de Madrid  
Facultad de Informática  
Departamento de Arquitectura de Computadores y Automática

Trabajo de Investigación Master

# **Sistema de identificación y seguimiento de superficies geográficas basadas en UAV con visión artificial**

por

**Pablo Lanillos Pradas**

Directores: José Jaime Ruz Ortiz, Gonzalo Pajares Martinsanz

Madrid, 2008



\*\*\*\*\* Dedicado \*\*\*\*\*

*a H.W*



---

# Índice general

---

<b>Índice general</b>	<b>I</b>
<b>1 Introducción</b>	<b>3</b>
1.1. Antecedentes . . . . .	4
1.2. Objetivo del presente trabajo: Identificación y seguimiento de superficies geográficas . . . . .	5
<b>2 Búsqueda</b>	<b>9</b>
2.1. Introducción . . . . .	9
2.1.1. Objetivo . . . . .	9
2.1.2. Escenario . . . . .	10
2.2. Mapas Probabilísticos . . . . .	10
2.2.1. Mapas estáticos . . . . .	11
2.2.2. Mapas dinámicos . . . . .	11
2.3. Estrategia de máxima probabilidad . . . . .	13
2.4. Estrategia de contención de frontera . . . . .	14
2.4.1. Definición de frontera . . . . .	14
2.4.2. Midiendo el rendimiento . . . . .	16
2.4.3. Algoritmo de contención . . . . .	18
2.4.4. Experimento del algoritmo de contención . . . . .	21
2.5. Estrategia Híbrida . . . . .	21
2.5.1. Experimentos . . . . .	25
<b>3 Identificación</b>	<b>31</b>
3.1. Generador de waypoints de alto nivel . . . . .	32
3.2. Seguimiento de contorno . . . . .	33
3.2.1. Detección de Contorno . . . . .	36
3.2.2. Reconstrucción de la superficie . . . . .	37
3.2.3. Experimentos . . . . .	38

<b>4 Seguimiento y Monitorización</b>	<b>41</b>
4.1. Introducción . . . . .	41
4.2. Definición como un problema MILP . . . . .	42
4.3. Smart Shape Tracking . . . . .	46
4.3.1. Algoritmo SST . . . . .	47
4.3.2. Particiones de la superficie usando una elipse . . . . .	47
4.4. Estudio de la altura y la huella del sensor . . . . .	50
4.5. Política de asignación para múltiples UAVs . . . . .	52
4.5.1. Definiciones . . . . .	53
4.5.2. Contorno de variación lenta en el tiempo . . . . .	54
4.5.3. Mancha de petróleo . . . . .	55
4.6. SST con información local . . . . .	56
<b>5 Conclusiones</b>	<b>59</b>
5.1. Aportaciones . . . . .	59
5.2. Trabajo Futuro . . . . .	60
<b>Bibliografía</b>	<b>63</b>
<b>A Seguimiento activo, aplicación para superficies sobredimensionadas</b>	<b>65</b>
A.1. Aplicación sobre manchas de petróleo . . . . .	65
<b>B Implementación del sistema</b>	<b>67</b>
B.1. Etapa de filtrado . . . . .	68
B.2. Generador de trayectoria de alto nivel . . . . .	69
B.3. Controlador del quadrimotor . . . . .	69
B.3.1. Control de la altitud por visión . . . . .	70
B.3.2. Control del balanceo . . . . .	71
B.3.3. Control del cabeceo . . . . .	71
B.3.4. Control de posición para seguimiento . . . . .	71
B.3.5. Control de rumbo . . . . .	72
<b>Índice de figuras</b>	<b>73</b>

---

# Agradecimientos

---

Durante la elaboración de este trabajo han contribuido una serie de personas que sin ellas no habría sido posible la finalización del mismo. En primer lugar me gustaría agradecer las aportaciones de Luca Bertucelli del departamento aeroespacial del MIT. Tanto en la estructuración y realización del proyecto como en las numerosas aportaciones en el campo de la búsqueda con incertidumbre su ayuda ha sido indispensable. También me gustaría agradecer a Till Junge, estudiante visitante del MIT, por su ayuda en los temas de modelado y en todas las aportaciones realizadas a la hora del café. Así como un verdadero apoyo moral durante la realización del trabajo. Al laboratorio aeroespacial del MIT por permitirme tenerme entre sus líneas y por la enorme acogida de todos sus colaboradores. Finalmente quiero agradecer la ayuda, que en la distancia, mis directores del trabajo de fin de master me han dado, tanto en las ideas como en la estructuración del trabajo. Por supuesto la labor administrativa de Kathryn Fisher secretaria del departamento aeroespacial también merece un agradecimiento.

Gracias a todos por haber sido parte de este trabajo y esto sin mencionar a todas las personas que me han seguido durante mi estancia en Estados Unidos.





## Capítulo 1

---

# Introducción

---

El desarrollo de la tecnología aeronáutica en el campo de los vehículos aéreos no tripulados (UAVs) ha abierto nuevas posibilidades de aplicación de las técnicas de procesamiento de imágenes tomadas en tiempo real desde estos vehículos. La presencia en el mercado de pequeños UAVs de bajo costo con capacidad para portar pequeñas cámaras de vídeo de alta resolución y de realizar despegue vertical con posibilidad de movimiento en cualquier dirección del espacio, hace posible abordar nuevos retos en el campo de la detección y seguimiento de determinadas situaciones de la realidad. Por ejemplo:

- Intervención y coordinación en catástrofes naturales: incendios, naufragios, manchas de petróleo en el mar, etc. (detección y seguimiento de incendios, detección y seguimiento de manchas de petróleo en el mar).
- Identificación de áreas de cultivo con determinadas anomalías
- La vigilancia de fronteras
- Inspección de infraestructuras como líneas eléctricas, redes de poliductos o redes de gasoductos
- Seguimiento y control del tráfico de vehículos terrestres
- Control de edificaciones en zonas urbanas
- Vigilancia aérea o marítima

Estas posibilidades se amplían cuando en lugar de un UAV utilizamos múltiples UAVs siguiendo una política coordinada de cooperación en la realización de la correspondiente tarea. Éstas posibilidades constituyen uno de los principales elementos motivadores para el presente trabajo de investigación, en el que nos

planteamos el problema de la búsqueda, identificación y seguimiento de áreas geográficas desde UAVs utilizando visión artificial.

## 1.1. Antecedentes

Este trabajo tiene sus orígenes en los trabajos previos desarrollados por el grupo de investigación ISCAR. Concretamente a través del proyecto con la empresa EADS-Casa al amparo del artículo 83 de la LOU titulado “Evaluación de técnicas de Inteligencia Artificial para toma de decisiones y optimización de estrategias” aplicadas al campo de la simulación para UAVs

Los trabajos anteriores tiene su continuación en el proyecto CICYT (DPI2006-15661-C02-01) actualmente en activo titulado “Plataforma de planificación, simulación y control para colaboración de múltiples vehículos autónomos marinos y aéreos”. Bajo la cobertura de este proyecto el autor de este trabajo posee una beca FPI, que ha posibilitado la estancia de los cuatro últimos meses en el MIT en USA.

Algunos trabajos han sido divulgados en el marco de la investigación y constituyen una fuente de referencia [PROL<sup>+</sup>08, RAPdlC07, RADlCP06]:

Los objetivos del proyecto DPI2006-15661-C02-01 son los siguientes:

1. Desarrollar una arquitectura (plataforma) software para la identificación y seguimiento de superficies geográficas mediante UAVs dotados de sistemas de percepción.
2. Realizar un estudio sobre el estado del arte de los diferentes subproblemas.
3. Analizar y desarrollar algoritmos de “Asignación de Tareas y Planificación de Caminos” para UAVs. Esta tarea la podemos descomponer en las siguientes subtareas.
  - Asignación y planificación automática de tareas para los UAVs
  - Generación automática de caminos para cada UAV en función de las tareas asignadas
  - Automatización de la toma de decisiones en la generación de caminos.
4. Desarrollo de modelos de entorno para desarrollar las diferentes misiones.
5. Estudiar y adaptar nuevos algoritmos para las aplicaciones previstas
6. Identificar nuevos problemas y abrir nuevas vías de investigación

## 1.2. Objetivo del presente trabajo: Identificación y seguimiento de superficies geográficas

Las aplicaciones citadas al principio del presente capítulo participan de una serie de características comunes que podemos resumir en cuatro:

1. Generación de una trayectoria que lleve el UAV a la zona de búsqueda.
2. Búsqueda del objetivo siguiendo trayectorias con máxima probabilidad de detección.
3. Identificación del contorno del objetivo
4. Seguimiento del objetivo y transmisión de su ubicación a una estación terrestre.

El objetivo vendrá identificado por el perímetro de un área geográfica que podrá ser estática o dinámica. En el segundo caso nos referiremos siempre a objetivos que se desplazan a una velocidad compatible con la velocidad de los UAVs.

La generación de una trayectoria estática óptima que lleve el UAV a la zona de búsqueda, evitando obstáculos y zonas de riesgo, ya ha sido abordada por el grupo de investigación utilizando técnicas de programación matemática (MILP) y técnicas heurísticas (algoritmo A\* y genéticos). En el presente trabajo se han analizado los otros tres problemas: Búsqueda, identificación y seguimiento (ver figura 1.1), problemas ya planteados en la literatura como Justin et Al. y Eric W. Frew et Al. argumentan en [CF05] y [Fre04] respectivamente. Desde el punto de vista de nuestro problema en cuestión se puede dividir en tres bloques que se complementan:

1. **Búsqueda** de una trayectoria que recorra una zona buscando puntos que acumulen la máxima probabilidad de encuentro del contorno. La trayectoria planificada se interrumpirá cuando se detecte el contorno (criterio de detección). Se utilizará una función espacial de probabilidad de encuentro del contorno. El criterio de detección será la identificación por primera vez de una sección del contorno. La aportación en este tema es un estudio de técnicas de búsqueda de objetos estáticos bajo incertidumbre y búsquedas de objetivos dinámicos; la creación de un algoritmo de contención utilizando contornos; y el desarrollo de una estrategia de búsqueda híbrida donde se tiene en cuenta la contención y la búsqueda de máxima probabilidad. Aparte se introduce la posibilidad de usar técnicas de contención para la monitorización de superficies geográficas.

2. **Identificación** del objetivo y extracción del contorno. Para la identificación se necesita conocer las propiedades del objeto que se busca. En el caso concreto de una mancha de petróleo, las propiedades de área, perímetro y posición (contorno) son necesarias. La aportación en este problema es la especialización de algoritmos de procesamiento de imágenes para identificar superficies geográficas aplicado a una mancha de petróleo y el desarrollo un generador trayectoria de alto nivel que permita el seguimiento de contorno de forma horaria. Para su posterior monitorización se introduce una estrategia para ajustar una forma elipsoidal o poligonal.
3. **Seguimiento o monitorización** del objetivo. Dependiendo del objetivo, es necesaria la monitorización del mismo. Por ejemplo en misiones de contención de fuego la monitorización es imprescindible para la coordinación de los recursos de contención y para la toma de decisiones. En este tema la aportación se centra en el desarrollo de un nuevo algoritmo para la monitorización de superficies sobredimensionadas por medio de un enjambre de agentes.

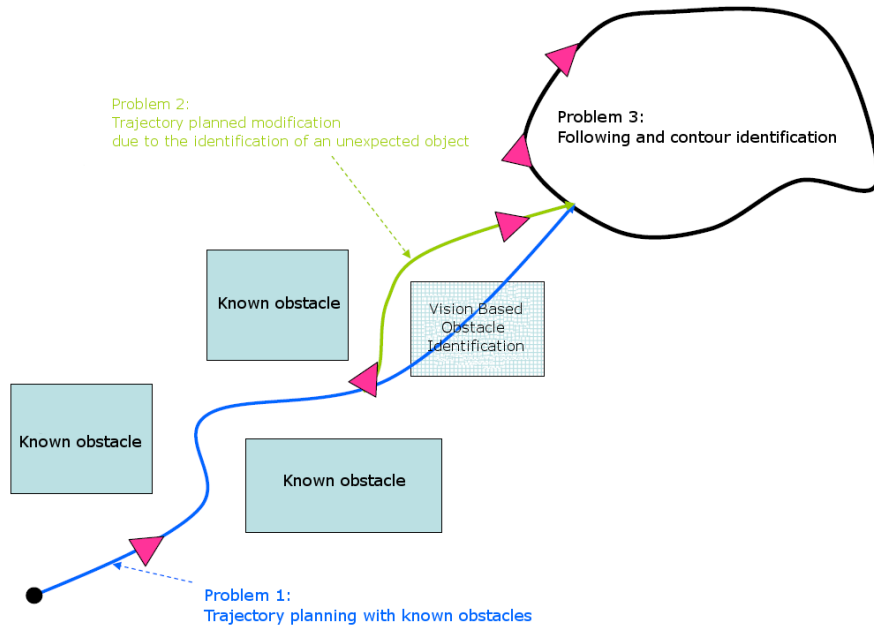


Figura 1.1: El primer subproblema es encontrar la trayectoria que maximice la probabilidad de encontrar el objetivo. El segundo subproblema es esquivar obstáculos inesperados, el tercer subproblema incluye la identificación y el seguimiento

El sistema tiene que poder adaptarse a cada una de las situaciones teniendo siempre en cuenta la seguridad del camino, es decir evitar las colisiones. El

seguimiento del objetivo dependerá de cada problema en cuestión. Por ejemplo en monitorización de fuego D. W. Casbeer et Al. [CBM<sup>+</sup>05], es importante hacer un seguimiento constante del objetivo para poder orientar a los servicios terrestres.

Dentro de este trabajo se abordan los subproblemas de búsqueda, identificación y seguimiento. Además se propone una arquitectura que permita la identificación de superficies terrestres sobre UAVs. El trabajo se organiza de forma que en el capítulo 2 se aborda el problema de la búsqueda de objetivos dinámicos con incertidumbre, que se corresponde con el subproblema 1. El capítulo 3 se centra en la identificación de superficies terrestres sobredimensionadas, que corresponde con el subproblema 3. Y Finalmente en el capítulo 4 se trata la monitorización y seguimiento de superficies, que corresponde con el subproblema 4. Para cada uno de los subproblemas se muestran los resultados obtenidos mediante las diferentes estrategias de test utilizadas. En el apéndice B se muestra un diseño y implementación de la arquitectura que ha permitido la prueba de los algoritmos para la identificación y seguimiento de superficies geográficas.



## Capítulo 2

---

# Búsqueda

---

En este capítulo exploramos el problema de la búsqueda desde una perspectiva general, no limitada a unas áreas geográficas. En el apéndice A se muestran dos ejemplos del uso de los algoritmos presentados a superficies geográficas.

### 2.1. Introducción

La búsqueda cooperativa ha sido estudiada en los últimos ocho años [BH06, WSZ05, YMP02, YMP04]. Realizaremos una revisión de las estrategias utilizadas para búsqueda estadística y se aborda un nuevo concepto para objetivos dinámicos utilizando la frontera de la probabilidad. Extendemos [BH06] incorporando seguimiento de contorno, definiendo el contorno o frontera como los últimos lugares donde las probabilidades del objetivo son propagadas debido al movimiento de este. Mientras que en [BH06] se habla sobre un problema de colocación de los UAVs, aquí el concepto de búsqueda toma mayor importancia. La predicción basada en cadenas de Markov proporcionará un incremento de probabilidades positivas en las zonas de la frontera, lo que significa que la probabilidad de encontrar al objetivo en las fronteras es probable y por lo tanto intentaremos que el objeto no escape. Es decir una estrategia de contención. Esta estrategia también puede proporcionar soluciones a vigilancia de fronteras con incertidumbre, reconstrucción de entornos, seguimiento de contornos sobredimensionados e identificación de superficies geográficas.

#### 2.1.1. Objetivo

La idea es buscar la frontera de las posiciones donde el objetivo podría moverse adelantándonos mediante predicción, e intentar realizar una contención de éste. El objetivo de este capítulo es estudiar el uso de una estrategia híbrida que

toma en consideración la búsqueda de máxima probabilidad (búsqueda de pico) y la frontera del mapa de probabilidad. Se usa el mapa de probabilidad como la información de la búsqueda y se utiliza como una función de aprendizaje para conducir las decisiones de los UAVs. La estrategia de frontera se utiliza para contener el objetivo y la búsqueda de pico es para objetivos más estáticos o que mueven en una zona uniforme. El gradiente del mapa de probabilidad actual y las probabilidades predichas son utilizadas para calcular la frontera en el siguiente paso temporal. El diseño del algoritmo de búsqueda está basado en la formulación de un controlador PD, con un termino derivativo que es la frontera, y un término proporcional que son los picos.

### 2.1.2. Escenario

El escenario es básicamente una región poblada por objetivos que necesitan identificación por algunos agentes (UAV). Los objetivos son dinámicos, así que se mueven sobre el entorno. El entorno es dividido en celdas  $M \times N$  con el tamaño del alcance del sensor del UAV. La decisión de la trayectoria depende de que celda deberá observar el UAV en el siguiente paso temporal. Cada celda contiene la información necesaria para la búsqueda como una matriz  $M \in R^{M \times N}$ . El movimiento del objetivo está definido por una matriz de transición y la velocidad para moverse de una celda a otra. Cada celda de la matriz de transición significa  $A_{ji} \doteq P\{X_{k+1} = j | X_k = i\}$ , así que la probabilidad del objetivo para saltar de una celda a otra es conocida por el UAV, pero no la posición real. El mapa de probabilidad  $\mathbf{P} \in R^{MN}$  es la información del entorno para el movimiento del objetivo.

## 2.2. Mapas Probabilísticos

Antes de entrar en detalles sobre la búsqueda es necesario definir como interpretar la información. Para su mejor entendimiento vamos a suponer en principio que los objetivos son estáticos. En la literatura podemos encontrar bastante información [YMP04]. Luego se extenderá a objetivos dinámicos [BH06], es decir, que pueden modificar su posición. No existe casi literatura concerniente a la búsqueda de objetivos dinámicos con incertidumbre. La principal fuente de información son artículos relacionados con cadenas de Markov. La idea principal es que no sabemos exactamente donde se encuentra el objetivo, pero tenemos una certeza de donde puede estar. Para modelar esa certeza se usarán mapas probabilísticos. Desde el punto de vista continuo  $P^i(x, y)$  es la función que nos proporciona la probabilidad de encontrar el objetivo  $i$  en la posición en el espacio  $x, y$ . Dado que tenemos un área donde el sensor del UAV puede actuar, podemos discretizar el entorno en celdas del tamaño del alcance del sensor. Al discretizar se podrá interpretar como cadenas de Markov (solamente se tiene en cuenta el estado anterior)



y podrán ser aplicadas para objetivos dinámicos. Así, podemos pensar en una matriz  $P$  que representa el entorno discretizado. Cada posición de la matriz determinará la probabilidad de encontrar el objetivo en esa región. Así la matriz queda definida en 2.1. Un valor de cero representa que es imposible encontrar lo que se busca en esa celda. En caso contrario, es decir, un 1, implica que la certeza de encontrar el objetivo en ese lugar es máxima. Como se introdujo en la definición del escenario, la matriz o mapa probabilístico es de tamaño  $M \times N$ .

$$P^i(i, j) = Pr(i, j), \text{ donde } \sum_{i,j} Pr(i, j) = 1 \quad (2.1)$$

La complejidad de cálculo es  $O(cte)$  que es el tiempo que se tarda en acceder a la posición en la matriz. El tamaño de la matriz viene dado por el disco de acción del sensor (cámara) y por el tamaño de la región de búsqueda (Ej. para un radio de acción de 100 metros del sensor y una región de  $1 \text{ km}^2$ , la matriz  $P$  tendrá un tamaño de  $10 \times 10$ ).

### 2.2.1. Mapas estáticos

Un mapa estático implica que no existe ningún movimiento de los objetivos. El mapa probabilístico sólo será modificado por la acción del sensor. Para ello se introduce el concepto del mapa temporal  $P_k^i$ . Para cada instante  $k$  existe una matriz de probabilidades. La predicción de encontrar el objetivo en el instante  $k$  está descrita así:

$$P_{k|k} = \hat{P}_{k|k} \quad (2.2)$$

Sea la acción  $X$  del sensor una función determinista que decide si hay presencia o no del objetivo, el mapa de probabilidades será modificado por la acción detección, actualizando el valor en esa celda y normalizando la matriz.

$$P_j = P_{ij}\delta(X = 1) + (1 - P_{ij})\delta(X = 0) \quad (2.3)$$

### 2.2.2. Mapas dinámicos

Con objetivos dinámicos es necesario tener en cuenta la dinámica del objeto que queremos identificar. El mapa probabilístico toma un significado parecido, pero este mapa determinará la probabilidad del objetivo para moverse desde la celda  $i$  a la celda  $j$ . Según el escenario definido anteriormente, algunos objetivos dinámicos deben ser identificados por el UAV en una región determinada. El movimiento probabilístico del objetivo (matriz de transición  $A$ ) y las características del UAV son conocidas. El controlador de búsqueda conduce las decisiones del UAV para detectar los objetivos utilizando el entorno como una función de aprendizaje que será actualizada cada vez que se realiza una observación. El objeto se moverá de acuerdo con su capacidad de movimiento y el objetivo es encontrar el

máximo número de objetos posibles en el menor tiempo posible. El entorno discretizado está compuesto por: el mapa de probabilidad para cada objetivo  $P_k^i$ ; el estado de los UAVs  $u_k^i$ ; la frontera predicha para las probabilidades en el siguiente paso temporal  $\Delta P_k^i$ .

- $A \in \mathbf{R}^{MN, MN} = \alpha_{ij}$  Es la matriz de transición del objetivo para el entorno  $M \times N$ . Representa la probabilidad de que el objetivo se mueva a la celda  $i \in 1, \dots, MN$  en el instante  $k + 1$  desde la celda  $j \in 1, \dots, MN$  en el instante  $k$ .
- $\mathbf{P}_k^i$  Es el mapa de probabilidad, si el entorno se divide en celdas en el instante  $k$  para cada objetivo  $i$ . El paso de predicción en el instante  $k + 1$  es calculado como:

$$\mathbf{P}_{k+1|k} = A\mathbf{P}_{k|k} \quad (2.4)$$

Teniendo en cuenta las medidas  $Y$  de detección tomadas por el sensor  $P_y = P_{ij}\delta(X=1) + (1-P_{ij})\delta(X=0)$ , y usando un vector en vez una matriz para representar el mapa probabilístico, cada valor nuevo de la celda será definido por la siguiente ecuación:

$$P_i(Y_{k+1|k}) = \sum_{j=1}^N (Pr(i_{k+q} | j_k) P_j(Y_k | k)) \quad (2.5)$$

$$= \sum_{j=1}^N \beta_{ij} P_j(Y_{k|k}), \quad \sum_{i=1}^N \beta_{ij} = 1 \quad (2.6)$$

- $\Delta \mathbf{P}_k^i$  Es el gradiente de la predicción del objetivo  $i$  en el instante  $k$  y es utilizado para definir la frontera como las aristas de la región de probabilidad. Utilizando Eq. 2.4 el gradiente de la predicción está definido como:

$$\Delta \mathbf{P}_{k+1}^i = (A\mathbf{P}_k^i) - \mathbf{P}_k^i \quad (2.7)$$

- $\mathbf{P}_k^c$  Es el mapa de probabilidad global, que es una combinación lineal de  $\mathbf{P}_k^i$  como es mostrado en Eq. 2.8.

$$\mathbf{P}_k^c = \sum \beta_i \mathbf{P}_k^i, \quad \sum_i^n \beta_i = 1 \quad (2.8)$$

- $\Delta \mathbf{P}_k^c$  Es el gradiente global de la predicción. Análogamente a la Eq. 2.7 pero utilizando el mapa global de probabilidad, podemos derivar la formula como:

$$\Delta \mathbf{P}_{k+1}^c = \sum \beta_i \Delta \mathbf{P}_k^i, \quad \sum_i^n \beta_i = 1 \quad (2.9)$$

## 2.3. Estrategia de máxima probabilidad

La estrategia de máxima probabilidad intenta que el UAV acumule la máxima probabilidad de encontrar al objetivo en su trayectoria. Teniendo el movimiento probabilístico del objeto a detectar podemos optimizar el camino para conseguir la máxima recompensa que es encontrar el objetivo. Esta estrategia puede también ser llamada estrategia de pico. Como podemos observar en la figura 2.1, la región de probabilidad forma unas curvas de nivel donde el centro en este caso representa la máxima probabilidad. El camino buscado irá al epicentro de las probabilidades o pico de la función probabilidad. Con la toma de muestras y el paso del tiempo conseguimos que el mapa se allane en las zonas visitadas y que la probabilidad crezca en las zonas no visitadas, siempre de acuerdo con la matriz de transición que define la dinámica del objetivo.

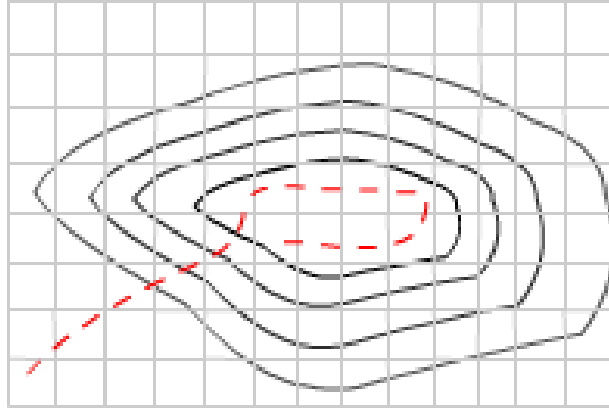


Figura 2.1: Trayectoria de máxima acumulación de probabilidad

Las decisiones que tome el UAV influirán en la trayectoria y por lo tanto en la acumulación de probabilidades de encontrar al objetivo.

$$k \rightarrow \{P^k\}$$

$$k + 1 \rightarrow \{P^k + P^{k+1}\}$$

$$k + 2 \rightarrow \{P^k + P^{k+1} + P^{k+2}\}$$

La ecuación a optimizar bajo la condición que la detección o no detección no afecta sobre el computo es la siguiente:

$$J_{i,j}^k = \arg \max \sum_k^{\infty} P^k \quad (2.10)$$

El algoritmo para encontrar la solución del camino de máxima acumulación es explicado posteriormente en la estrategia híbrida.

## 2.4. Estrategia de contención de frontera

Dada alguna probabilidad de movimiento la estrategia de contención en vez de visitar la celda que maximiza la probabilidad de encontrar el objetivo, intenta contener la propagación del objetivo visitando las celdas que es más probable que éste se mueva, anteponiéndose a los movimientos. Dividimos el entorno en celdas  $M \times N$  y colocamos el objetivo con su mapa de probabilidad asociado en la celda  $i, j$ . Si tenemos una distribución uniforme para la propagación del objetivo, dinámica completa para ir a cualquiera de las celdas adyacentes y asumimos que no existen observaciones o medidas desde el UAV, las celdas donde el objetivo puede ser encontrado en el conjunto 8-conectado de celdas de  $i, j$ . Este conjunto de celdas es exactamente la frontera de las probabilidades. Utilizando la predicción como la manera para conocer que celdas el objetivo posiblemente visitará en el siguiente instante de tiempo, la frontera o contorno de las probabilidades es definido con el paso de la predicción.

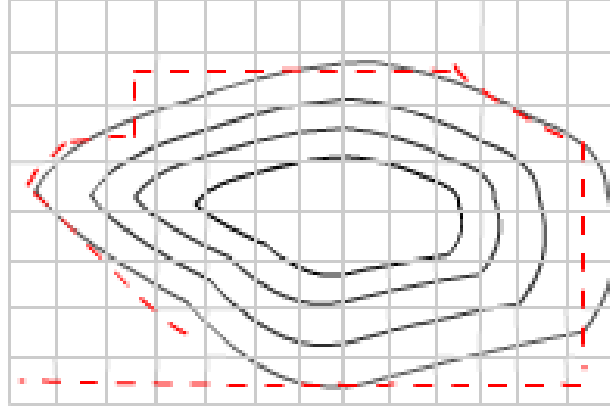


Figura 2.2: Trayectoria de contención

### 2.4.1. Definición de frontera

- $\partial \mathbf{B}_k^i$  Es la frontera local en el instante  $k$  del objetivo  $i$ . Como el gradiente calculará también pesos negativos debido al decremento de la probabilidad, es necesario combinar el mapa de probabilidad  $\mathbf{P}_k^i$  y el gradiente predicho  $\Delta \mathbf{P}_{k+1}^i$ . La frontera local son las celdas donde en el instante  $k$  son cero y en el instante  $k+1$  son mayores que cero. Así que el objetivo podría ser encontrado en esas celdas en el siguiente instante de tiempo.

$$\partial \mathbf{B}_{k+1}^i \doteq (\mathbf{P}_k^i = 0 \wedge \Delta \mathbf{P}_{k+1}^i > 0) \quad (2.11)$$

- $\partial \mathbf{B}_k^c$  Es la frontera global, una combinación lineal de  $P_k^c$ . Podemos derivar de la misma forma que se realiza en la Eq. 2.11.

$$\partial \mathbf{B}_{k+1}^c \doteq (\mathbf{P}_k^c = 0 \wedge \Delta \mathbf{P}_{k+1}^c > 0) \quad (2.12)$$

También podemos definir el contorno global usando el cierre y el interior y calculando la intersección para computar la frontera Eq. 2.14. El proceso para su cálculo es el siguiente:

Sea  $\overline{S}_k^i$  el cierre  $\partial \mathbf{B}_k^i$  y  $S_k^{i0}$  el interior de  $\partial \mathbf{B}_k^i$ , cada frontera local es definida como:

$$\partial \mathbf{B}_k^i = \overline{S}_k^i \setminus S_k^{i0} \quad (2.13)$$

La región definida por  $\partial \mathbf{B}_k^i$ , será  $R_k^i = \overline{S}_k^i$ . Si la intersección de las regiones es calculada  $\mathbf{I}_k = \bigcap_{i=1}^n R_k^i$  el contorno global es definido como sigue:

$$\partial \mathbf{B}_k^c = \overline{\mathbf{I}_k} \setminus \mathbf{I}_k^0 \quad (2.14)$$

La frontera puede ser resuelta como un problema de detección de bordes ampliamente estudiado en segmentación de imágenes. Así que análogamente a una formulación con imágenes, definimos la frontera como una matriz binaria  $B \in [0, 1]^{M \times N}$  donde

$$B[i, j] = \begin{cases} 1 & \text{if } B_{i,j} \in \partial \mathbf{B}, \\ 0 & \text{if } B_{i,j} \notin \partial \mathbf{B}. \end{cases}$$

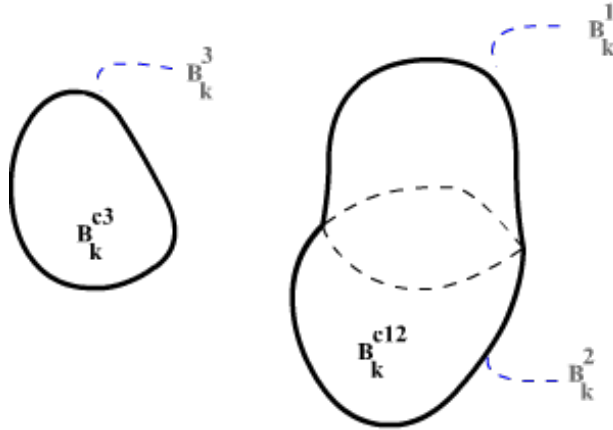


Figura 2.3: Fronteras combinadas.  $B_k^{c12} = \alpha(B_k^1 + B_k^2)$  and  $B_k^{c3} = B_k^3$ . La frontera global es una combinación lineal:  $B_k^c = \alpha(B_k^{c12} + B_k^{c3})$

Las aristas o bordes pueden ser encontradas utilizando el gradiente, que en este caso es la diferencia entre las probabilidades predichas en el instante  $k +$

1 y las probabilidades actuales en  $k$ . No todos los bordes son necesarios para extraer la forma exterior de la región como en procesamiento de imágenes, sólo las transiciones entre cero y no cero en el paso de predicción. Dependiendo del numero de objetivos dos tipos de fronteras ha sido definidas: Frontera local, asociada con cada objetivo (la frontera de la región descrita por los posibles lugares que el objetivo puede ir); Frontera global, que combina todos las fronteras locales.

Tenemos la unión de las fronteras pero es imposible extraer información como el número de zonas de probabilidad que hay en el entorno. Hay que pensar que tenemos fronteras probabilísticas que algunas intersecan y otras permanecen aisladas. Para recuperar esa información de los mapas de probabilidad, una segmentación de la imagen binaria es necesaria. La frontera son los puntos de la curva paramétrica  $(x(t), y(t))$  para cada región, así un algoritmo de seguimiento de contorno es necesario.(e.j. para cada punto de la frontera queremos encontrar el punto siguiente y el punto anterior). Existen numerosos métodos propuestos en la literatura de visión por computador. Como tenemos un mapa binario de la frontera global cualquier método será suficientemente rápido para procesamiento en tiempo real. Utilizando este método podemos extraer las propiedades de cada región formadas por la propagación de las probabilidades del objetivo.

### Fronteras para objetivos rápidos

Con dinámicas de gran velocidad el paso de propagación se realiza muchas veces y produce que el contorno se disperse demasiado rápido. La definición de la frontera en Eq. 2.12 es una frontera del 100 %, que significa que toda la probabilidad está contenida en su interior. En vez de propagar todas las celdas en la predicción del contorno, un umbral  $\delta$  es utilizado para reducir la dispersión. El nuevo contorno es definido como la frontera que contiene el  $R$  % de las probabilidades (Eq.2.15). Celdas con baja probabilidad no son tomadas en consideración restringiendo la masa de probabilidad incluso cuando el objetivo es rápido.

$$\partial \mathbf{B}_k^c R \% = (\mathbf{P}_k^c > \delta \wedge \Delta \mathbf{P}_{k+1}^{c'} > 0) \quad (2.15)$$

$$\Delta \mathbf{P}_{k+1}^{c'} = A P_k^{c'} - P_k^{c'} \text{ donde } P_k^{c'} = P_k^c > \delta \quad (2.16)$$

### 2.4.2. Midiendo el rendimiento

En esta sección se muestra una forma de medir la eficiencia y rendimiento del controlador de frontera. Si las observaciones son siempre que el objetivo no es encontrado, algunas medidas son extraídas para conocer el rendimiento de la estrategia en diferentes escenarios. Estas medidas nos indican si el controlador puede manejar la situación. La probabilidad de éxito y el número de UAVs necesarios para el éxito de la misión dependiendo de la dinámica del objetivo como

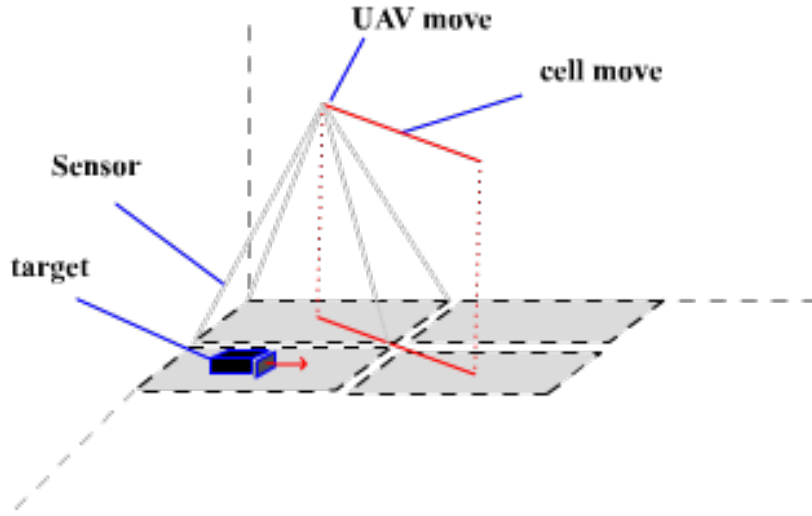


Figura 2.4: La huella del sensor define el tamaño de la celda. Los segundos necesarios para que el objetivo cambie de una celda a otra y la velocidad del UAV para observar la celda  $i$  definen el parámetro  $kupdate$ .

la tasa de actualización de la propagación versus las celdas visitadas por el UAV (Sea el tamaño de la celda  $l \times l$  y definiendo la dinámica del objetivo como el tiempo en media para cambiar de una celda a otra,  $kupdate = \frac{l/v_{target}}{l/v_{UAV} + t_{Observation}}$  que puede ser simplificado si  $t_{Observation} \approx 0$  como  $kupdate = \frac{v_{UAV}}{v_{Target}}$ ). Si el parámetro  $kupdate$  incrementa el UAV tiene más tiempo para visitar celdas. En caso opuesto el objetivo es más rápido, así que el número de celdas que el UAV puede visitar se reduce. (e.j. Asumiendo que la velocidad media del objeto es 10 mps y el la huella del sensor del UAV puede capturar 100 m en una altura determinada de vuelo a partir del índice focal de la cámara. Definimos la celda con 100 m de longitud de lado. El objetivo podría cambiar a cualquier celda con un mínimo tiempo de 5 segundos, así usando un  $kupdate$  de 8 significa que la velocidad del UAV deberá ser de 288 mps).

Principalmente el controlador de contorno tendrá éxito en dos casos. El primer caso cuando el UAV puede visitar todas las celdas de frontera antes del siguiente paso de propagación. El segundo caso ocurre cuando al calcular el contorno global un conjunto vacío es devuelto. Esto ocurrirá cuando no exista ninguna fuente para las probabilidades.

- Definimos éxito como el número de celdas de frontera visitadas dividida por el número total de celdas de frontera.

$$success = \# \frac{|\partial \mathbf{B}|_k}{\partial \mathbf{B}_{TOT}}$$

- Probabilidad de éxito,  $P(success)$ , muestra como el controlador trabaja de acuerdo con el entorno y los parámetros del objetivo.

$$P(success) = \frac{1}{total} \sum_i success_i$$

Tomando en consideración el objetivo, el éxito del controlador depende exclusivamente en cuantos objetos son encontrados. Así, esta primera definición de éxito se refiere al caso peor.

El limite del éxito para la contención puede ser calculado analíticamente buscando la intersección de la caja del UAV definida por su dinámica y la propagación de la frontera formada por el objetivo.  $\Delta T$  será el tiempo que se necesita para llegar a la frontera propagada. (e.j para una una distribución uniforme en un entorno  $MN$  existe una expansión de  $2(N + M + 2K) - 4$ , ver figura 2.5).

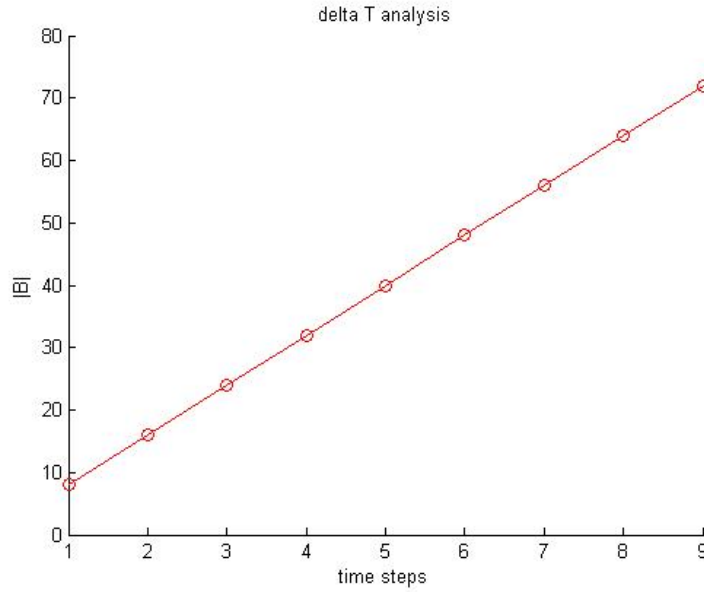


Figura 2.5: Análisis del tiempo para la contención de la frontera

### 2.4.3. Algoritmo de contención

El algoritmo de contención se divide en tres pasos importantes explicados en la figura 2.6, donde se producen las observaciones de los UAV, luego se efectúa el planeamiento del camino y finalmente si se necesita la propagación del objetivo se realiza, dando paso de nuevo a la fase de observación.



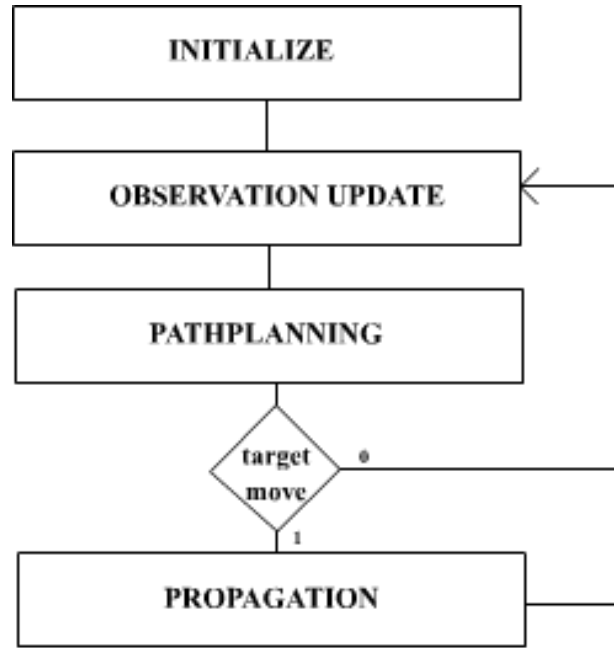


Figura 2.6: Algoritmo de contención, diagrama de flujo abstracto para la arquitectura del sistema

### Actualización por Observación

La actualización del mapa depende de la observación tomada por una cámara modelada de forma determinística, donde la salida es 1 cuando el objetivo es detectado y 0 cuando no es detectado. (Eq. 2.3). El conjunto de observaciones entre cada paso de propagación conducirá la búsqueda. Las medidas restringirán la región de la posición probable del objetivo. El mapa de probabilidad y el mapa de contorno son actualizados con la medida.

### Planeamiento del Camino

Para decidir la trayectoria que el UAV seguirá, partimos desde estas hipótesis extraídas del entorno:

- Una estrategia de seguimiento de contorno no puede ser utilizada si no existe frontera o si ésta está lejos.
- Si existe alguna frontera cerca del UAV, éste tiene que seguirlo de una alguna forma “óptima”.

Sea  $u_k$  la posición del UAV en el instante  $k$  y  $d_{k+1}$  el destino del UAV, se presenta un algoritmo 2.4.3 que tiene en cuenta las hipótesis mencionadas. Si no existe frontera el UAV intentará acercarse a la celda con máxima probabilidad de encontrar el objetivo (búsqueda del pico “greedy”). En el caso de que una frontera

este presente pero lejos, el UAV irá a la celda más cercana de frontera (contención de frontera “greedy”). Es posible ir a la celda con máximo gradiente pero es muy importante llegar cuanto antes posible a la masa de probabilidad para reducir la incertidumbre, así la estrategia para ir a celda más cercana proporciona una mejor tasa de éxito. Finalmente si el UAV tiene la frontera cerca, seguirá el contorno de una manera “óptima” utilizando un planeamiento del camino de paso adelante.

---

**algorithm 1** Algoritmo de Path Planning
 

---

```

  if  $\sum_{i \in \text{adj}(u_k)} \mathbf{B}_k^c = 0$ 
    if  $\partial \mathbf{B}_k^c = \emptyset$ 
      GreedyPeakFinding
       $d_{k+1} = c_q$  where  $\mathbf{P}_k^c(c_q) = \max \mathbf{P}_k^c$ 
      Determine  $c_r$  as the adjacent cell of  $u_k$  that is in the direction of  $d_{k+1}$  using the angle  $\alpha$  as  $\arctan u_k d_{k+1j}, u_k d_{k+1i}$ 
       $u_{k+1} = c_r$ 
    else
      GreedyBoundaryContainment
       $d_{k+1} = c_q$  where  $\Delta \mathbf{P}_k^c(c_q) = \max \Delta \mathbf{P}_k^c$ 
      Determine  $c_r$  in an analog way as before
       $u_{k+1} = c_r$ 
    end
  else
     $d_{k+1} \dots d_{k+n} = n - \text{stepaheadsearch}(\partial \mathbf{B}_k^c, \Delta \mathbf{P}_k)$ 
     $u_{k+1} = d_{k+1}$ 
  end

```

---

El algoritmo paso adelante desarrollado es un  $\alpha\beta$  ramificación y poda que encuentra el camino de longitud  $n$  ( $Q_{k+1}^1, \dots, Q_{k+1}^n$ ) que maximiza en  $n$  pasos la máxima diferencia de las probabilidades  $\mathbf{P}_k^c$ , que es lo mismo que encontrar el camino que maximiza  $\Delta \mathbf{P}_k^c$ . Debido a la gran purga de los estados por que las posibles posiciones es restringida por el contorno, la velocidad del algoritmo es aproximadamente el rendimiento de un algoritmo voraz. Normalmente sólo hay dos ramas abiertas en la búsqueda. Así que será  $2 * (n \text{ pasos adelante})$ .

Sea  $W \in R^{M_0 N_0}$  la ventana de celdas donde  $n * 2 + 1$  es el número de celdas de cada lado, rellena por los valores de  $\Delta \mathbf{P}_k$ , que necesitamos para mirar  $n$  pasos más adelante, y  $\text{adj}(i)$  los posibles estados que el UAV puede llegar desde la celda  $i$  una definición recursiva del algoritmo es proporcionada. La función para el caso recursivo está basado en una función multiobjetivo.

(base case) $\text{pr}(i) = W(i)$	$\longleftarrow n = 0$
(rec. case)	$\longleftarrow n > 0$

$$\text{pr}(i) = W(i) + \gamma^n \left( \max_{\substack{a \in \text{adj}(i) \\ a \in \partial \mathbf{B}_k^c}} \text{pr}(a) \right) - \alpha^n d(i, a)$$

El coste del camino es  $\sum_{i=1}^n Q^i$  y es máximo en  $W \in R^{M_0 N_0}$ . Las restricciones para conducir la poda son: los estados válidos son celdas de frontera; el camino no puede contener estados repetidos; los estados utilizados por otros UAVs no pueden estar en el camino.

### Paso de propagación

En cada actualización del objetivo el paso de propagación es realizado en este orden:

1.  $\mathbf{P}_{k+1}^c = A\mathbf{P}_k^c$
2.  $\partial \mathbf{B}_{k+1}^c = (A\mathbf{P}_k^c = 0 \wedge \Delta \mathbf{P}_{k+1}^c \geq 0)$

#### 2.4.4. Experimento del algoritmo de contención

El experimento posee estos parámetros: MN=100, NUA=1, NTARGETS=1, n steps ahead=4, maxtime=100. En la figura 2.7 se puede observar como la índice de paso de propagación (*kupdate*) afecta a la probabilidad de éxito. Como era de esperar, altos índices de pasos de propagación implican menos éxito en media. Si el controlador falla, el tiempo computado es igual al tiempo máximo de la búsqueda. Existe un límite donde si seguimos incrementando *kupdate* no existirá mejor rendimiento. Esto demuestra empíricamente el valor derivado analíticamente en 2.4 para la contención de la frontera.

## 2.5. Estrategia Híbrida

El objetivo es encontrar objetos de una manera óptima utilizando la información proporcionada por el conocimiento del entorno. Los mapas de probabilidad  $\mathbf{P}_k^c$  y  $\Delta \mathbf{P}_k^c$  conducirán las decisiones de los UAVs. El estado del entorno está definido como  $x(t) = \{P(t), \Delta P(t), u(t)\}$ . Las decisiones de los UAVs están representadas por  $u(t) = [u_1(t), \dots, u_N(t)]$ . Así la decisión en el instante  $t+1$  es una función del estado del entorno en el instante  $t$  y del estado del entorno en el instante  $t+1$ , es decir, el estado actual, las decisiones y las nuevas observaciones hechas por los sensores.

Queremos maximizar una función de recompensa  $G(x(t), u(t))$  y minimizar el coste  $D(x(t), u(t))$ . La función de coste tiene en consideración que la distancia para los movimientos diagonales son mayores que la de los otros movimientos.

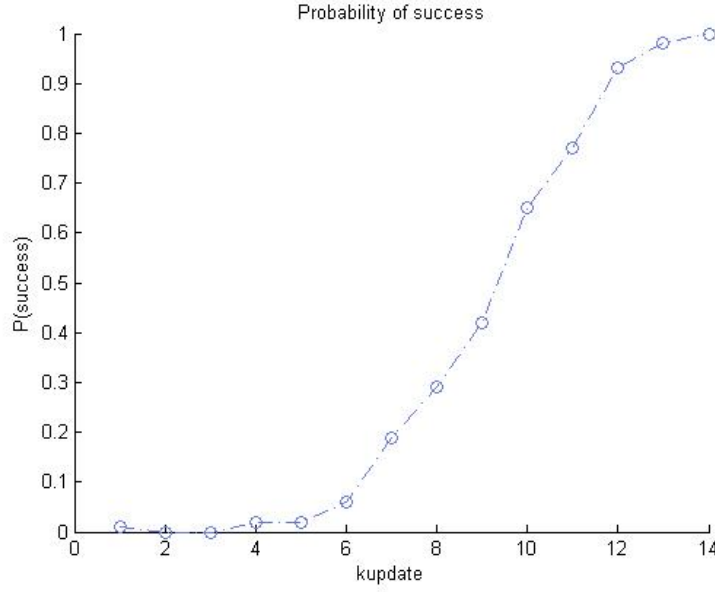


Figura 2.7: Simulación de Montecarlo para 100 ejecuciones, 1 UAV y un objetivo en un entorno de 10x10 celdas

Así, movimientos diagonales son penalizados. Esto es necesario porque la frontera está definida por formas cuadradas y queremos observar las esquinas en vez de dejarlas aisladas, para que no se propaguen de nuevo.

$$D(x(t), u(t)) = \sum_{i=1}^M d(u_i(t)) \quad (2.17)$$

La función  $d$  es la distancia del UAV  $i$  al estado designado  $u_i(t)$ . El problema de búsqueda se diseña como una definición de controlador óptimo donde se tiene un termino proporcional  $\mathbf{P}_k^c$  y un termino derivativo  $\Delta\mathbf{P}_k^c$ . También podemos definirlo como un controlador híbrido piramidal donde, dependiendo del entorno, uno u otro controlador conducirá la trayectoria del UAV. El controlador de búsqueda está descrito en 2.26. Con este tipo de enfoque el controlador se comporta como una búsqueda informada multiobjetivo. Así, el controlador trabajará como contención de frontera en el caso de que exista frontera y como una búsqueda de pico en otros casos. Los pesos de  $K_p$  y  $K_d$  modificarán la importancia de cada controlador.

$$u^* = K_p \mathbf{P}_k^c + K_d \Delta\mathbf{P}_k^c \quad (2.18)$$

Queremos encontrar el camino  $Q^n$  donde  $\sum_{i=1}^n Q^i$  es máxima. Por fuerza bruta, la solución puede ser computada creando todas las posibles combinaciones de subconjuntos de longitud  $n$ . El número de posibles soluciones viene dado por

2.19.

$$(C_n^k) = \binom{n}{k} = \frac{n!}{k!(n-k)!} \quad (2.19)$$

La explosión combinatoria para grandes  $n, k$  puede ser tratada precomputando los subconjuntos off-line. Este cálculo puede ser acelerado dividiéndolo en subproblemas. Calcular el subconjunto para cada nodo y después mezclarlos, tomando sólo los caminos válidos.

$$\text{subset}(Q^n) = \text{subset}(Q^{n/2}) \cup \text{subset}(Q^{n/2})$$

La complejidad computacional online está dada por la complejidad de encontrar el máximo de los subconjuntos máx  $\text{subset}(Q^n)$ . El coste de almacenamiento es demasiado grande incluso si tenemos en cuenta que para ir de un nodo a otro sólo existen 8 posibilidades como máximo.

La estrategia recursiva propuesta en la contención de contorno puede ser utilizada tomando para cada comportamiento un tabla de costes  $W1, W2$ . Esto no es práctico en tiempo real con un tamaño de camino de más de 6, debido a la explosión combinatoria del árbol recursivo. Esta estrategia puede ser acelerada utilizando una tabla indexada (DP) donde los valores ya calculados de longitud  $s$  para cada nodo son almacenados.

$$\begin{aligned} \text{(base case)} \quad \text{pr}(i) &= W1(i) + W2(i) && \longleftarrow n = 0 \\ \text{(rec. case)} &&& \longleftarrow n > 0 \\ \text{pr}(i) &= W1(i) + W2(i) + \gamma^n \left( \max_{a \in \text{adj}(i)} \text{pr}(a) \right) - \alpha^n d(i, a) \end{aligned}$$

Para mostrar la complejidad del problema podemos abstraerlo a teoría de grafos. Un digrafo puede ser construido utilizando las celdas como nodos, así para un entorno  $M \times N$  se tiene  $MN$  nodos. La solución óptima para un camino de longitud  $MN$  es encontrar el ciclo hamiltoniano de máximo coste. Esto es un caso especial del problema del viajante de comercio (TSP), que significa que es NP-completo. La complejidad es la de un FNP para un problema de decisión, así que es un problema NP-duro. La programación lineal puede ser utilizada para obtener una solución como un problema TSP. Pero por ser un entorno discretizado donde tenemos que tomar decisiones, el enfoque de resolverlo con las ecuaciones de Bellman 2.20 como un problema de decisión de Markov, permitirá incluso añadir incertidumbres en el movimiento del UAV.

$$V^*(s) = G(s) + \max_a \gamma \sum_{s'} P(s' | s, a) V^*(s') \quad (2.20)$$

Sea  $G(i, j, k)$  la recompensa para ir de  $i$  a  $j$  en el instante  $k$  la recompensa infinita es

$$\max E_j \left\{ \sum_{K=0}^{\infty} \gamma^K \lambda^K G(i, j, k) \right\} \quad (2.21)$$

Utilizando la detección del sensor 2.3 y 2.22 implica  $G(i, j) = P_{ij}$

$$G(i, j, k) = G(i, j) = \begin{cases} 1, \\ 0 \end{cases} \quad (2.22)$$

En cada paso con una actualización de propagación de índice alto, la recompensa acumulada para el camino con medida 0 es:

$$\begin{aligned} k &\rightarrow P^k \\ k+1 &\rightarrow \{P^k + \lambda P^{k+1}\} \\ k+2 &\rightarrow \{P^k + \lambda P^{k+1} + \lambda^2 P^{k+1}\} \end{aligned}$$

Así la optimización de horizonte finito e infinito está definida por esta ecuación:

$$\max \left\{ \sum_{k=0}^{MN} \lambda^k P_{ij} \right\} \approx \max \left\{ \sum_{k=0}^t \lambda^k P_{ij} + J_{\text{cost-to-go}} \right\} \quad (2.23)$$

Debido al factor de descuento temporal, no existe optimalidad probada en el final del camino seguido por el UAV, así la decisión sólo será localmente óptima para un entorno específico en el instante  $k$ . La acción de la medida, debido a la normalización, hace que la decisión no sea óptima. (e.j. si en el instante  $k$  la mejor solución es hacer  $\{0, 1, 2\}$ , en el instante  $k+1$ , dependiendo del factor de descuento, la solución óptima podría ser  $\{1, 2\}$  o  $\{2, 1\}$ ). La condición que hace que exista este salto es la siguiente:

En el instante  $k$  tenemos una política que construye este camino:  $\{0, 1, 2\}$

$$p_0 + \lambda(P_1) + \lambda^2(P_2)$$

Si la medida es una no detección del objetivo ( $z = 0$ ) la normalización implicada es:  $P_1 = \Delta P_1$  y  $P_2 = \Delta P_2$  donde  $\Delta = \frac{1}{P_1 + P_2}$ . Las posibles decisiones en el instante  $k+1$  son:

$$\lambda \Delta P_1 + \lambda^2 \Delta P_2$$

$$\lambda \Delta P_2 + \lambda^2 \Delta P_1$$

La condición para cambiar la decisión habiendo tomado la celda número 0 es:

$$\lambda \Delta P_2 + \lambda^2 \Delta P_1 > \lambda P_1 + \lambda^2 P_2 \quad (2.24)$$

Así se necesita replanificación, dependiendo de  $\Delta$  y  $\lambda$ . La conclusión es que un óptimo local es calculado. La decisión será óptima hasta el siguiente paso de propagación. Para objetivos estáticos el óptimo local se convierte en óptimo global.

Utilizando las ecuaciones de Bellman definimos la función de recompensa como una formulación de controlador PD derivado de Eq. 2.18.

$$G(i, j) = K_p \mathbf{P}_k^c(i, j) + K_d \Delta \mathbf{P}_k^c(i, j) \quad (2.25)$$

Finalmente la política de búsqueda óptima se encuentra maximizando la recompensa menos la función  $D(x(t), u(t))$ . La solución óptima viene dada por la Eq. 2.26.

$$u^* = \arg \max_{i,j} \left( \sum \lambda G^k(i, j) \right) - J(x(t), u(t)) \quad (2.26)$$

Una matriz de transición  $T$  y una matriz de recompensa  $R$  para el UAV y el entorno es necesario. La matriz de transición está definida como la probabilidad de ir del estado  $i$  a  $j$  ejecutando la acción  $a$ :  $T = P_{ij}^a = \{j \mid i, a\}$ . Sea  $C_{ij}$  la representación de la recompensa del entorno, la matriz de recompensa está definida por:  $R_i^a = \{C_{ij} \mid i, a \rightarrow j\}$ .

Utilizando iteración de valor y cuatro acciones que el UAV puede tomar en cada celda, la media de cómputo para un entorno de  $20 \times 20$  es  $\approx 1$  segundos (con una implementación en Matlab).

### 2.5.1. Experimentos

Se han realizado pruebas con las diferentes estrategias con objetivos dinámicos usando una matriz de transición de distribución aleatoria uniforme. Modificando los pesos  $K_p$  y  $K_d$  en la función de recompensa podemos cambiar del comportamiento de contención de frontera a búsqueda de pico pura. La figura 2.8 muestra el UAV usando una estrategia de contención. Este permanece en los márgenes de la masa de probabilidad intentando que ningún objetivo se escape. La figura 2.9 muestra la búsqueda de máxima probabilidad pura, es decir, la estrategia de pico. El UAV intenta alcanzar el centro de la masa de probabilidades, acumulando a su paso el máximo. En la figura 2.10 se muestra el caso híbrido donde se intentan aunar las dos estrategias. El UAV intenta cubrir el contorno o acercarse al pico dependiendo de los cambios en el entorno.

Los resultados para el caso híbrido para una distribución uniforme aleatoria no mejora en media la búsqueda de máxima probabilidad. Este resultado puede ser observada en la figura 2.11. El caso híbrido con  $K_d = 1, K_p = 1$  es siempre un límite superior para el éxito y un límite inferior para el tiempo de éxito. El uso del gradiente de la probabilidad  $\Delta P$  y el mapa de probabilidad  $P$  como información conjunta produce, con altos valores de  $K_d$ , comportamientos de cambio brusco siguiendo siempre la misma trayectoria. Esto es debido a que las celdas que se han visitado recientemente tienen un alto gradiente de cambio. Para que el sistema híbrido funcione correctamente se deberán eliminar los comportamientos de "flip-back".

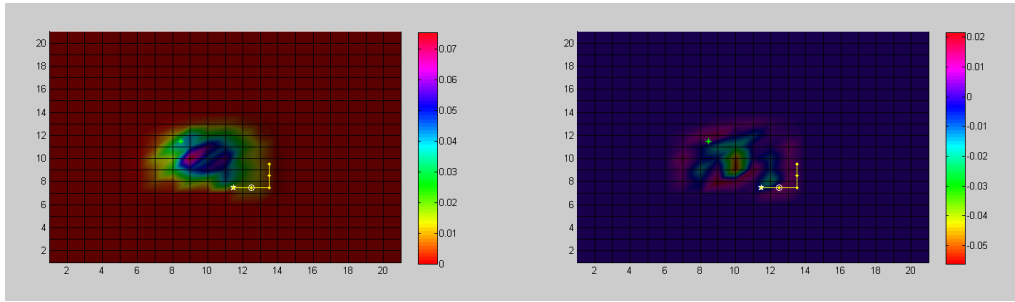


Figura 2.8: La figura muestra el comportamiento de contención para una función de recompensa con  $K_p = 0, K_d = 1$ . En la derecha se encuentra el mapa de probabilidad del objetivo y en la derecha el gradiente entre la predicción y el mapa de probabilidad  $AP - P$ . El camino seguido por el UAV es la frontera de la región de probabilidad.

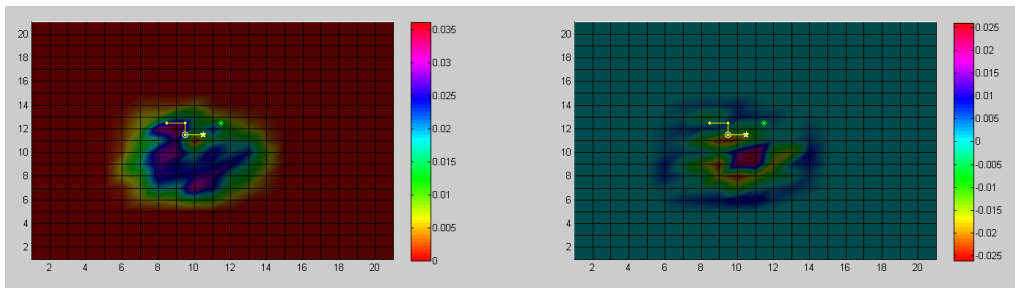


Figura 2.9: La figura muestra el comportamiento de pico con una función de recompensa con  $K_p = 1, K_d = 0$ . En la derecha se encuentra el mapa de probabilidad del objetivo y en la derecha el gradiente entre la predicción y el mapa de probabilidad  $AP - P$ . El camino seguido por el UAV intentará acumular el máximo de las probabilidades acercándose al pico.

### Matriz de transición para carreteras

En estos experimentos se desea comprobar el uso de las estrategias para entornos más realistas como pueden ser las carreteras. Para la realización de los experimentos se modificará la matriz de transición de los objetivos para que tengan el comportamiento de un coche circulando por la carretera. También se desea comprobar como afecta la bifurcación de caminos en la búsqueda.

Dos tipos de carreteras son definidas por la matriz de transición: carreteras de dos direcciones, el objetivo puede ir hacia atrás; carreteras de una dirección, el objetivo nunca puede ir hacia atrás. En la figura 2.12, se muestra el mapa de probabilidad formado por este tipo de matriz de transición. En este experimento el objetivo intentará alcanzar una posición antes de que el UAV le detecte. El patrón utilizado es un rombo donde el objetivo puede tomar dos caminos diferentes para alcanzar el destino. Con una carretera de una dirección la probabilidad



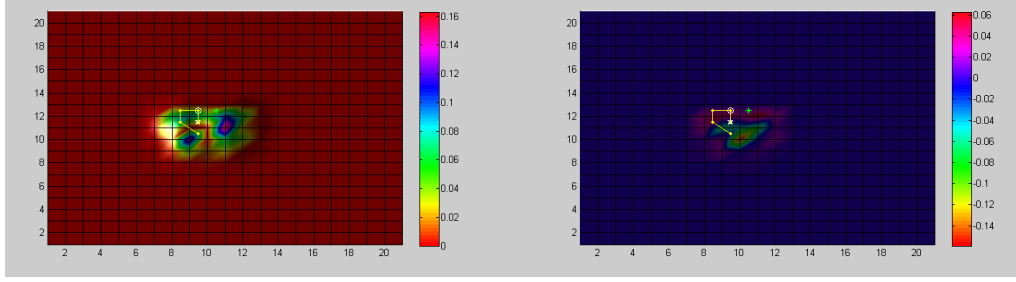


Figura 2.10: La figura muestra el comportamiento híbrido para una función de recompensa con  $K_d = 10, K_p = 1$ . El camino elegido por el UAV tiene en cuenta la frontera modificando la trayectoria hacia la frontera de la región.

de éxito para  $kupdate = 1, 2$  se muestra en la figura 2.13. Esta simulación de montecarlo con 100 ejecuciones por cada  $kupdate$  muestra la probabilidad de éxito dependiendo de los parámetros elegidos para  $K_d$  y  $K_p$ . El patrón para la carretera hace que el objetivo elija aleatoriamente uno de los dos caminos para llegar al destino.

### Cooperación de UAV con diferentes controladores híbridos

En este experimento se refleja el uso de un enjambre de UAVs con diferentes pesos para la contención y la búsqueda de máxima probabilidad. En la figura 2.14 se describe la posibilidad de utilizar múltiples UAVs con diferentes comportamientos sobre un objetivo con una matriz de transición de distribución uniforme aleatoria. La línea roja representa la probabilidad de éxito para dos UAVs con búsqueda de máxima probabilidad y la línea negra muestra la probabilidad de éxito donde uno de los UAV está en modo búsqueda pura del pico y el otro está realizando búsqueda híbrida.

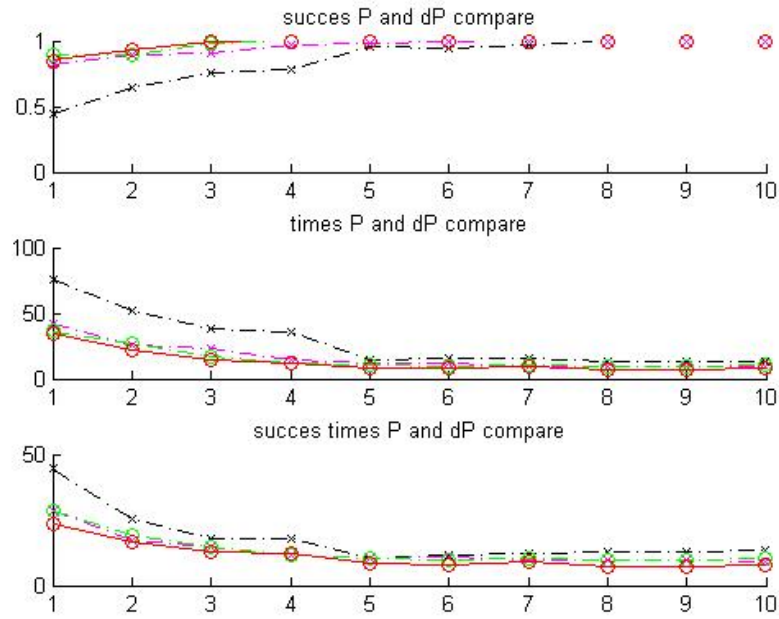


Figura 2.11: Simulación de Montecarlo con 100 ejecuciones, 1 UAV y un objetivo. La figura superior muestra la probabilidad de éxito (eje “y” es  $P(success)$  y el eje “x” es el parámetro  $kupdate$  o índice de velocidad del objetivo), La figura del medio representa la media de tiempo para cada ejecución. La figura inferior muestra la media en tiempo de encontrar el objeto. Para las figura del medio y la inferior el eje y es el tiempo en pasos requeridos y el eje x es la tasa de  $kupdate$ . El color representa los siguientes pesos:  $K_d = 1, K_p = 0$ , negro;  $K_d = 10, K_p = 1$ , morado;  $K_d = 0, K_p = 1$ , verde;  $K_d = 1, K_p = 1$ , rojo

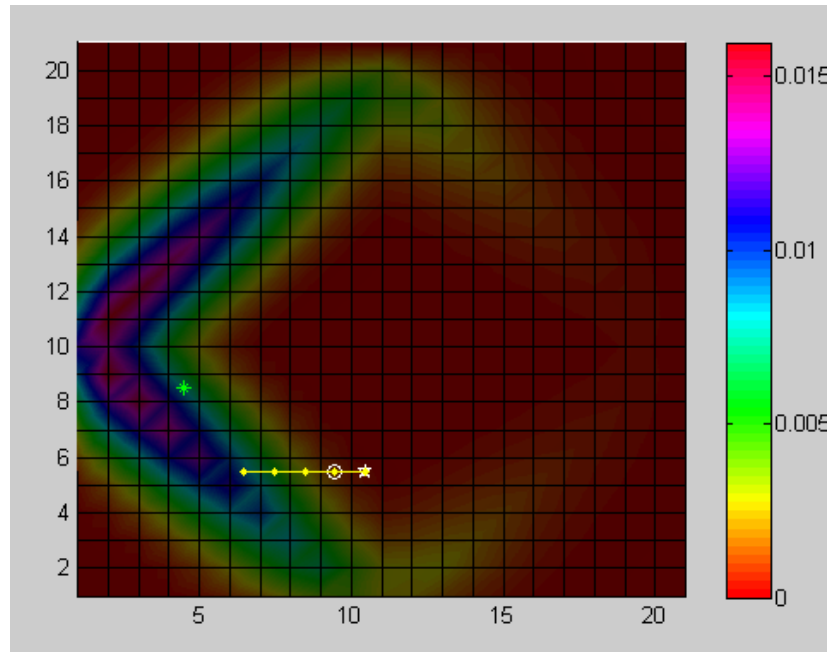


Figura 2.12: Mapa de probabilidades formado por un patrón de rombo como una carretera bifurcada.

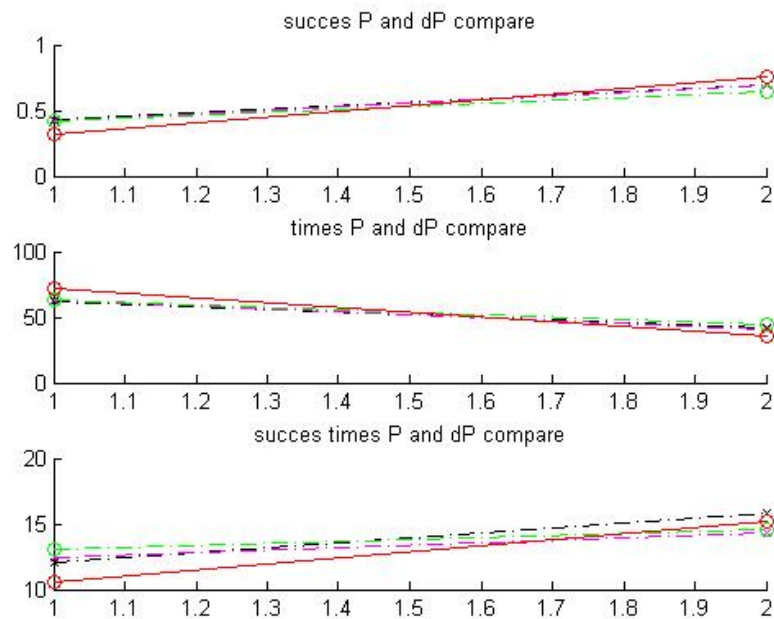


Figura 2.13: Esto es una simulación de montecarlo con 100 ejecuciones, La línea roja representa la pura búsqueda de pico, y los otros, diferentes configuraciones de caso híbrido. La línea negra es  $K_d = 1, K_p = 1$

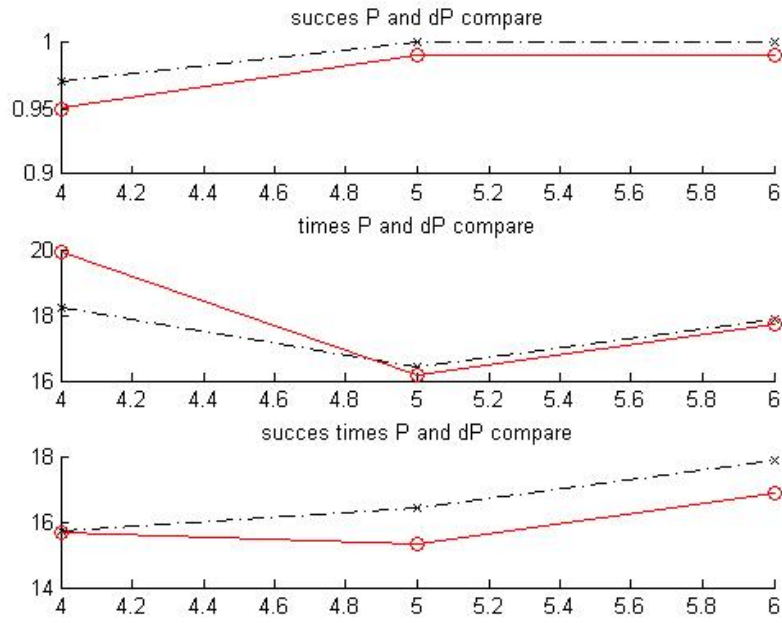


Figura 2.14: Simulación de Montecarlo con 100 ejecuciones, se muestran dos tipos de configuración de búsqueda con un objetivo a buscar. La línea negra muestra la configuración mixta con dos UAVs, donde uno está realizando búsqueda pura del pico y el otro está realizando una búsqueda híbrida con pesos  $K_d = 1, K_p = 1$ . La línea roja es una configuración con 2 UAVs utilizando sólo búsqueda de pico  $K_d = 0, K_p = 1$ .

## Capítulo 3

# Identificación

En este capítulo se explica como abordar el subproblema 3 en la identificación y seguimiento de superficies geográficas. La principal aportación en el capítulo es la especialización de algoritmos de procesamiento de imágenes para identificar superficies geográficas aplicado a una mancha de petróleo, y el desarrollo de un generador de puntos de trayectoria de alto nivel que permita flexibilidad a la hora de tomar decisiones.

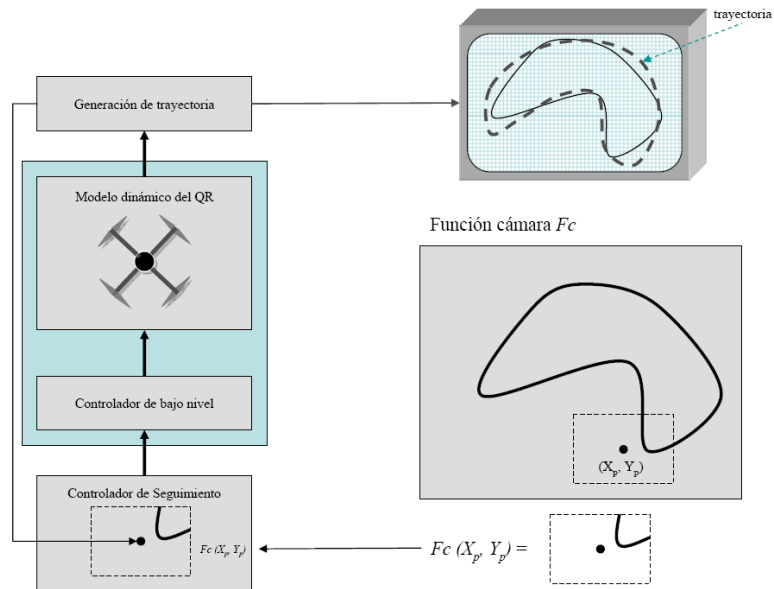


Figura 3.1: Para la identificación vamos a usar una arquitectura flexible que permita la inclusión de la cámara y que gracias a un generador de trayectorias podamos ejecutar diferentes estrategias

### 3.1. Generador de waypoints de alto nivel

El generador de trayectorias (GTAN) va a decidir que puntos deberá seguir el UAV para completar la misión asignada. Dependiendo del estado del UAV y de la información recibida por medio de los sensores, el generador de trayectorias puede estar en diferentes modos de ejecución o estados. El vector de estado del GTAN se compone de dos flags:

$$GTAN(modulo, t)$$

El flag *modulo* indica en que tarea se encuentra y el flag *t* indica si esta tarea se ha terminado o no. Se puede ver como un autómata finito, o como la secuencia de acciones de alto nivel de un agente, figura 3.2.

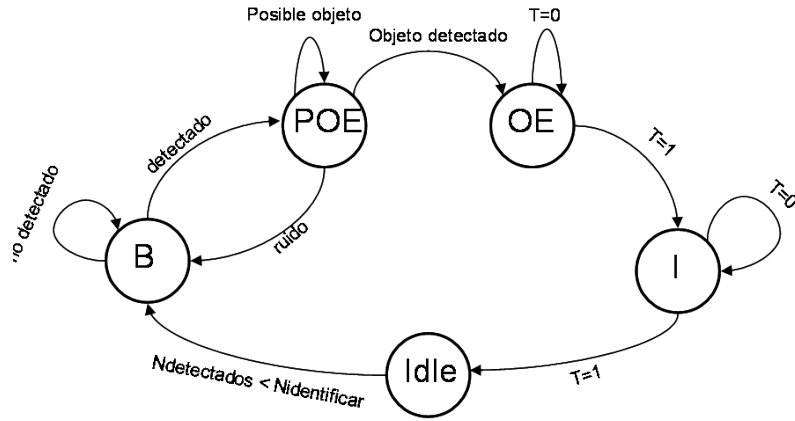


Figura 3.2: Diagrama de estados y transiciones del generador de trayectorias de alto nivel

1. **Modo búsqueda (B):** Los sensores no han detectado la superficie buscada, así que la trayectoria seguida por el UAV vendrá determinada por una estrategia de búsqueda como por ejemplo los algoritmos descritos en el capítulo 2.
2. **Modo Objeto-Encontrado (OE):** Cuando la certeza de la detección sobrepasa un determinado valor se establece que el objeto ha sido localizado y se procede a la colocación del UAV en la frontera de la superficie.
3. **Modo Identificación (I):** El generador de trayectorias calculará, dependiendo de la estrategia elegida, los puntos necesarios que se tienen que visitar para la identificación de la superficie. (Ej. Seguimiento del contorno en sentido horario.)
4. **Modo Seguimiento:** Todas las tareas de identificación han sido ejecutadas (Ej. no existen más objetos que identificar) y se procede a la monitorización.

Para pasar de la etapa de búsqueda a la de identificación, tenemos que estar seguros de que los sensores nos están dando la información correcta. Así tendremos un límite de probabilidad donde estaremos totalmente seguros de que hemos encontrado el objetivo. Es importante en nuestro caso, el filtrado de las imágenes capturadas por la cámara. La primera vez que encontremos un contorno que satisfaga las propiedades del objetivo teniendo en cuenta el campo de visión de la cámara, podremos pasar a la correcta identificación.

Dado que los objetivos pueden ser más grandes que el campo de detección del sensor, necesitamos una maniobra de aproximación. Para la maniobra de aproximación usaremos tanto el centro de masas de la imagen filtrada como la información que sabemos de la superficie en cuestión. Para manchas de petróleo en el mar es importante saber cuando se produjo la catástrofe para estimar el tamaño del contorno y así tener una base con la que comparar para extraer las dimensiones del objetivo.

Sea los puntos de la imagen filtrada cuyo valor es uno, el centro de masas será:

$$\overrightarrow{R_{CM}} = \frac{\sum(\vec{r}_{(i,j)}m_{(i,j)})}{\sum m_{(i,j)}}$$

Dado que el peso de cada punto tiene como valor uno por el uso de una imagen binaria, el centro de masas resulta:

$$\overrightarrow{R_{CM}} = \frac{\sum(\vec{r}_{(i,j)})}{\sum p_{\vec{r}}}$$

Una vez que nos hemos aproximado a la superficie, hay que identificarla. Existen numerosas formas de seguimiento del contorno. La propuesta por Susca et Al [SBM08], parece la más indicada para contornos dinámicos ya que distribuye la colmena de UAVs usando el método de serpiente de manera uniforme sobre la superficie. A continuación se presentan dos métodos que explotan el campo de visión de la cámara usando un único UAV. El primero encuentra la siguiente celda (siendo cada celda del tamaño del sensor) en el sentido de las agujas del reloj para recorrer todo el contorno de la superficie. El segundo intenta explotar la información que tenemos sobre el objetivo y con pocos puntos hace encuentra una forma que se ajusta a la superficie encontrada.

### 3.2. Seguimiento de contorno

La estrategia utilizada en este caso es el seguimiento de contorno en el sentido de las agujas del reloj es decir que geográficamente se recorrerá de la forma:  $\{N, NE, E, SE, S, SO, O, NO\}$ . Asumimos que nos encontramos sobre el contorno de la superficie que queremos identificar y que está sobredimensionada, el sensor no puede abarcar toda la superficie. El sensor utilizado es una cámara situada en el UAV haciendo un ángulo de  $90^\circ$  con el eje  $x$  e  $y$  del UAV. Una vez la

cámara captura la imagen, ésta queda almacenada en una matriz  $C(i, j) \in R^{M \times N}$  donde cada posición contiene un píxel. La imagen pasa de la cámara a un módulo de filtrado. Definimos la información filtrada como una matriz binaria  $D$  que representa la detección de la superficie. Dada la superficie  $\Omega$  se cumple:

$$D[i, j] = \begin{cases} 1 & \text{si } D_{i,j} \in \Omega, \\ 0 & \text{if } D_{i,j} \notin \Omega. \end{cases}$$

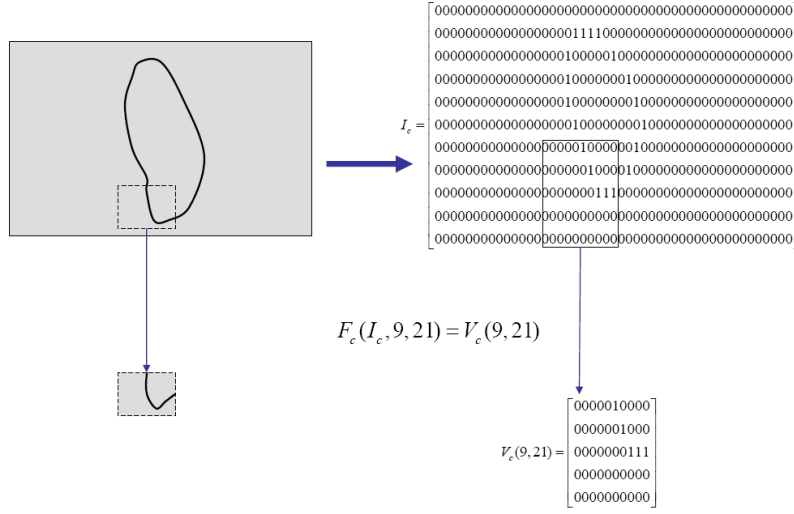


Figura 3.3: El sensor que se usará es una cámara colocada sobre el UAV enfocando al suelo es decir  $90^\circ$  con respecto al eje del UAV.

A continuación necesitamos el contorno o frontera de  $\Omega$  denotada como  $\partial B$  como se muestra en la figura 3.3. Ésta será otra matriz  $B$  donde:

$$B[i, j] = \begin{cases} 1 & \text{if } B_{i,j} \in \partial B, \\ 0 & \text{if } B_{i,j} \notin \partial B. \end{cases}$$

Las etapas son mostradas de forma conceptual como un modelo de tubería en la figura 3.4. La imagen es almacenada en una matriz de píxeles que posteriormente es filtrada por un detector binario y en la tercera etapa el contorno es extraído.



Figura 3.4: La imagen es almacenada en una matriz de píxeles que posteriormente es filtrada por un detector binario y finalmente el contorno es extraído



Posteriormente se explican diferentes formas de extraer el contorno y cuales son sus ventajas. Estas etapas pueden verse representadas como la función cámara  $F(x, y)$  que aparece en la figura 3.3.

Una vez determinado el contorno necesitamos saber a qué región nos debemos dirigir para capturar la siguiente imagen, véase la figura 3.6. Sea  $\partial B^\#$  la función que define el contorno de la superficie en la huella  $\#$  de la cámara y sea  $f^\#$  el polígono que define la huella, los puntos necesarios para decidir la siguiente región viene dada por la intersección de  $\partial B^\#$  y  $f^\#$ , y la dirección en el sentido horario que lleva el UAV.

$$\kappa = \partial B^\# \cap f^\# \quad (3.1)$$

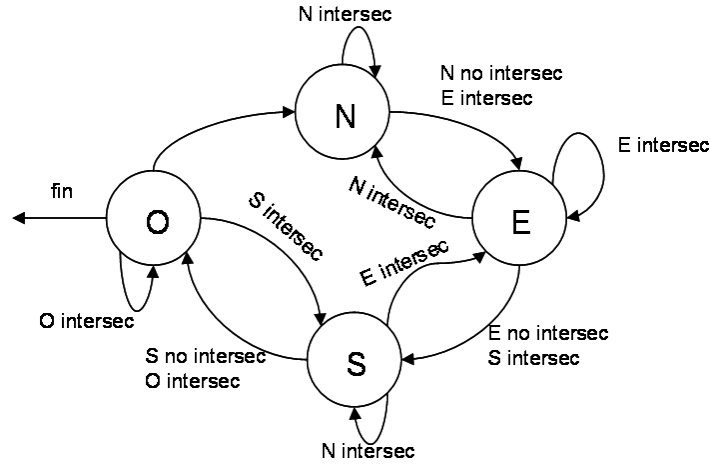


Figura 3.5: El esquema representa el diagrama de estados para recorrer en sentido  $\{N, E, S, O\}$  la superficie a identificar

Para elegir la dirección  $\{N, E, S, O\}$  se usará el diagrama de estados que aparece en la figura 3.5 (se comenzará en el estado Norte y dependiendo si hay intersecciones o no con el lado norte de la huella del sensor se pasa al estado siguiente). Las intersecciones  $\kappa$  pueden encontrarse en los lados del polinomio  $f^\#$ , es decir, en  $\{N, E, S, O\}$ . La función  $\partial B^\#$  puede ser aproximada mediante segmentos. Así, el problema reside en comprobar si alguno de los segmentos interseca entre sí. La complejidad de ese cómputo es de  $O(N \log N)$ , siendo  $N$  el número de segmentos. Dentro del conjunto de intersecciones sólo vamos a utilizar el máximo y el mínimo y para ello vamos a especificar un poco más en detalle  $\kappa$ .  $\kappa(x, y)$  son las coordenadas cartesianas que definen los puntos de intersección con la huella del sensor. Sea  $\kappa_x$  y  $\kappa_y$  el subconjunto de los valores de la  $x$  e  $y$  respectivamente en  $\kappa$ . El punto que generará la dirección es definido como sigue:

- Caso N:  $\kappa(x, y) \mid x = \max \kappa_x$

- Caso E:  $\kappa(x, y) \mid y = \max \kappa_y$
- Caso S:  $\kappa(x, y) \mid x = \min \kappa_x$
- Caso O:  $\kappa(x, y) \mid y = \min \kappa_y$

Ya tenemos la dirección  $\vec{v}$  a seguir y en este paso se puede optar por usar el punto extraído como siguiente waypoint o elegir como siguiente punto el centro de la región determinada por  $\vec{v}$ , como se indica en la figura 3.6

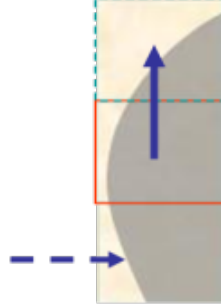


Figura 3.6: El espacio se divide en  $M \times N$  teniendo en cuenta el disco de detección de la cámara y la altura del UAV. La decisión de la trayectoria reside en elegir cual es la nueva celda

### 3.2.1. Detección de Contorno

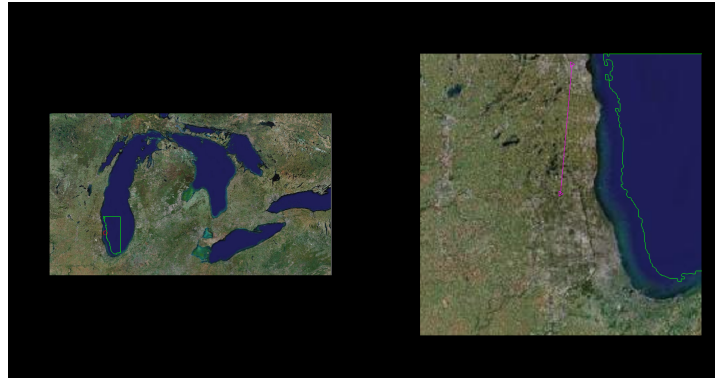


Figura 3.7: Mediante técnicas de visión artificial, se puede filtrar la imagen y extraer el contorno  $\partial B$  de la zona donde se encuentra el UAV

Existen numerosas técnicas de obtención de contorno, y en nuestro caso contamos con un mapa binario que representa la detección o no del objeto. En la figura 3.7 se muestra un ejemplo de detección de contorno. El más utilizado se basa en el seguimiento de contorno marcando los cambios de gradiente alto. Para el caso de una imagen binaria el procedimiento se explica en la figura 3.8. La

extracción del contorno se realiza mediante técnicas de seguimiento de los píxeles de tipo 4-conectado o 8-conectado.

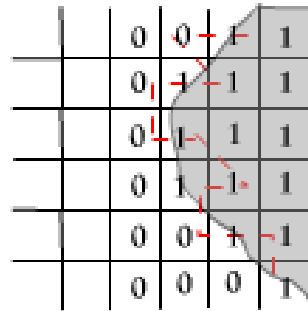


Figura 3.8: La extracción del contorno se puede realizar mediante técnicas de seguimiento del borde de tipo 4-conectado y 8-conectado

En este caso no necesitamos todos los puntos, sino determinados puntos que nos orienten a tomar la siguiente decisión. Una extracción de puntos de interés y la formación de una curva entre ellos sería suficiente. Dependiendo del rango del sensor y del tamaño de la superficie, el número de puntos necesarios variará. Si queremos definición a nivel de píxel, cualquier técnica de seguimiento de contorno servirá para la extracción. En la figura 3.9 se muestra cómo se puede extraer el contorno usando el algoritmo de Susan [SB95] para la extracción de puntos de interés. Se pueden unir los puntos de tal forma que tenemos segmentos que indican el contorno de la superficie.



Figura 3.9: No necesitamos todos los puntos de la frontera, sino sólo los que nos van a dar la orientación de hacia donde hay que dirigirse

### 3.2.2. Reconstrucción de la superficie

Después de la obtención de fotografías de la superficie necesitamos realizar la reconstrucción de la superficie (figura 3.10). Utilizando todas las imágenes adquiridas se genera una reconstrucción del objeto que se desea identificar. Cada imagen capturada tiene asociada las coordenadas geográficas de donde se tomó la foto. Asumiendo que la cámara siempre está a  $90^\circ$  con respecto al suelo, la unión

de las imágenes se realiza solapando cada una de ellas para formar una gran imagen  $I$ . Los píxeles situados en varias imágenes a la vez, es decir los píxeles solapados  $I(i, j) \in I_1, \dots, I_n$  se calcularán haciendo la media:

$$I(i)_{i \in I} = \sum_k^n I^k / n \quad (3.2)$$

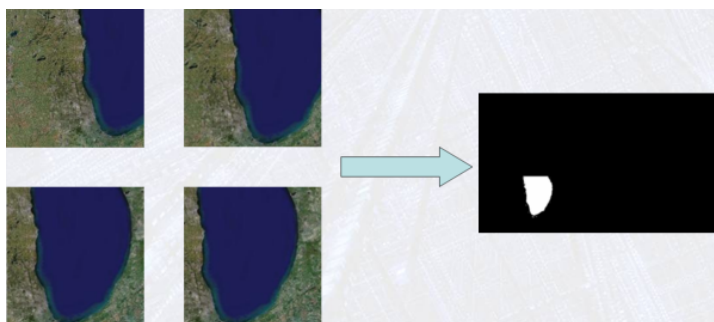


Figura 3.10: A partir de todas las imágenes adquiridas se generará una reconstrucción del objeto que se deseaba identificar

En el caso de fuerte ruido producido por la mala detección de los sensores o debido al movimiento del UAV, los ángulos del UAV toman una gran importancia y técnicas de correlación o matching más complejas deberán ser utilizadas.

### 3.2.3. Experimentos

En este experimento se probará el uso de la estrategia explicada en este capítulo reconociendo una superficie. Se pretende detectar dos objetivos que coincidan con unas características determinadas. En particular se pretende identificar el lago Michigan y el lago Huron. Para la simulación se ha utilizado una imagen extraída de la aplicación Google Earth, de los grandes lagos situados en Estados Unidos. Los lagos serán las superficies a identificar. En la figura 3.11 se muestra un instante de la simulación usando el generador de trayectorias de alto nivel, donde la imagen superior izquierda representa la región a explorar, la imagen superior derecha muestra la imagen capturada por la cámara del UAV, en la imagen inferior izquierda aparece el mapa de detección binario, es decir la imagen filtrada donde blanco implica detección y negro no detección, y finalmente en la imagen inferior derecha, encontramos el contorno extraído y las intersecciones con la huella del sensor.

La imagen reconstruida, a partir de las capturas de la simulación, se muestra en la figura 3.12. Podemos apreciar que se han recogido dos lagos, ya que el número de objetivos a identificar en la simulación era 2.

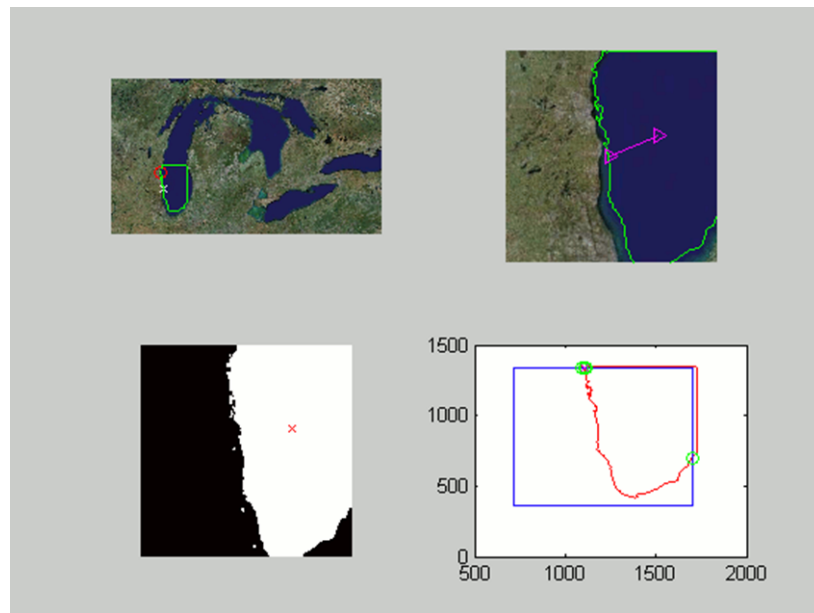


Figura 3.11: Captura de la simulación utilizando los puntos de interés para decidir la trayectoria



Figura 3.12: La imagen reconstruida representa toda la superficie identificada



## Capítulo 4

---

# Seguimiento y Monitorización

---

Este capítulo responde al subproblema identificado en 1.2 que es el seguimiento o monitorización de la superficie terrestre.

### 4.1. Introducción

En este capítulo se muestra una estrategia inteligente para el seguimiento de un contorno basada en el radio de acción del sensor. Una aproximación para tiempo real del problema de optimización será presentada como una buena forma de minimizar los recursos a la hora del seguimiento de contornos dinámicos. El objetivo es encontrar la trayectoria para seguir todo el contorno, incluso cuando el objeto está sobredimensionado (el área del sensor del agente es menor que el objeto a identificar) en comparación con el radio de acción del sensor. La idea es ajustar una forma al contorno y explotar las propiedades matemáticas de esa forma para encontrar el mínimo número de puntos de observación que cubran todos los puntos del contorno. Finalmente una estrategia para la cooperación de UAVs es diseñada para el uso de enjambres para el seguimiento del contorno.

La flexibilidad de los UAVs, nos permite el desarrollo de estrategias más eficientes e inteligentes para el seguimiento de objetos o contornos dinámicos. Existen multitud de enfoques para el seguimiento de objetos sobredimensionados. En [CF05] una búsqueda en espiral y un controlador de campo potencial es usado para seguir una mancha de petróleo. En [CBM<sup>+</sup>05] se habla sobre una estrategia de puntos equidistantes y un controlador de seguimiento en sentido horario para monitorizar fuegos. En [SBM08] Sara Susca et al. combinan el algoritmo de serpiente con vehículos sin piloto para distribuir los robots en un contorno que varía con el tiempo. Utilizan sensores de visión y una red inalámbrica para subir la información. Su algoritmo está basado en una aproximación óptima para cuerpos

convexos por polígonos. Aparte en [McL08], utilizan un conjunto o enjambre de robots para probar algoritmos como distribución uniforme o seguimiento de líder. Proponen una manera de evaluar el rendimiento de los algoritmos como: el ancho de banda el tiempo, la distancia, etc.

Nuevos algoritmos inteligentes o tareas complejas pueden ser desarrolladas hoy en día utilizando enjambres como una red de sensores. Estas estrategias siempre tienen en cuenta los recursos y el rendimiento. Este algoritmo que explota el área del sensor para reducir el consumo de recursos en un seguimiento de contornos de larga duración. La estrategia asume que siempre es posible encontrar una forma que se ajuste al contorno del objeto de interés. Se prueba mediante experimentos que la estrategia produce buenos resultados en tareas de larga duración.

## 4.2. Definición como un problema MILP

Este método parte del conocimiento del perímetro, cosa que no se conoce a priori. La solución del problema será la mínima trayectoria para realizar una identificación completa de la superficie geográfica buscada. La optimización tiene dos subproblemas: el número mínimo de puntos de observación que se necesitan y que la trayectoria para recorrer esos puntos sea mínima.

$$\min_{N, X_{op}} \sum_{i,j}^N (X_{op}^j - X_{op}^i)^2 \quad (4.1)$$

Para poder convertirlo en un problema lineal necesitamos hacer una aproximación de la distancia 1-norm o distancia Manhattan.

$$\min_{N, X_{op}} \sum_{i,j}^N |X_{op}^j - X_{op}^i| \quad (4.2)$$

Hay que tener en cuenta que estamos minimizando el perímetro del polígono formado por los puntos de observación y que por lo tanto tenemos que utilizar el absoluto ya que con la aproximación 1-norm las soluciones intentará acercarse al origen de coordenadas.

También necesitamos discretizar el contorno de la superficie, y comprobar que todos los puntos son observados. Luego las restricciones son:

Sea  $\partial B$  el contorno de la superficie, todo punto perteneciente a ese contorno tiene que estar contenido en el disco del sensor de alguno de los puntos de observación. Si definimos  $\#$  como la huella producida por el sensor en la superficie (la superficie que puede capturar la cámara), podemos utilizar una variable binaria para seleccionar en que huella se encuentra cada punto del contorno de la superficie.

$$\forall p \in \partial B, \delta p^\# \in \# \quad (4.3)$$



Debido a la posible geometría del objeto los puntos pueden estar contenidos en más de una huella.

$$\sum \delta p^\# \geq 1 \forall \# \quad (4.4)$$

Así que el problema definido en MILP es:

$$\begin{aligned} \min_{N, X_{op}} \quad & \sum_{i,j}^N |X_{op}^j - X_{op}^i| \\ \text{s.t} \quad & \forall p \in \partial B \delta p^\# \in \# \\ & \sum \delta p^\# \geq 1, \forall \# \end{aligned}$$

Por ejemplo, para un sensor radial con radio  $R$  la huella  $\#$  define esta restricción:

$$\forall p \in \partial B \mid \exists X_{op} \mid (X_{op} - p)^2 < R^2$$

La restricción lineal para el círculo viene dada por una aproximación en forma de polígono del círculo producido por el sensor [PROL<sup>+</sup>08]. Los puntos de observación obtenidos pueden verse en la figura 4.1.

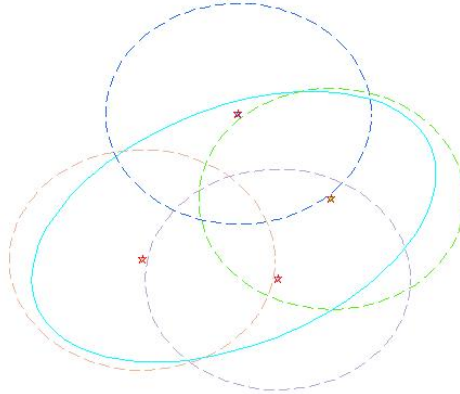


Figura 4.1: Resultado usando MILP y un sensor con huella circular

Dada la complejidad de cómputo para acelerar la búsqueda de la solución usamos un límite superior para el número de puntos de observación que nos proporcionará siempre una solución. El límite superior es calculado de la siguiente forma:

Sea la superficie  $\Omega$  y su función explícita  $z = f(x, y)$ , y la huella del sensor  $\#$  de radio  $R$ , podemos definir la caja contenedora (bounding box) con el máx  $f(x, y)$

y el mín  $f(x, y)$ , como la caja que contiene la superficie. Definimos el área de  $\Omega$  como

$$A(\Omega) = \int \int_{\Omega} \sqrt{1 + \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} dx dy$$

Y el perímetro de la mismo como:

$$P(\Omega) = \sum_{i,j}^N (p^j - p^i), \forall p \in \partial B$$

Así el límite superior del número de puntos  $NXop$  necesarios para cubrir el borde de la superficie es:

$$NXop = \min \left\{ \frac{bbox(\Omega)}{R}, \frac{A(\Omega)}{A(\#)}, \frac{P(\Omega)}{R} \right\} \quad (4.5)$$

Usando la aproximación 1-norm para la distancia obtenemos el polígono mínimo mostrado en la figura 4.2

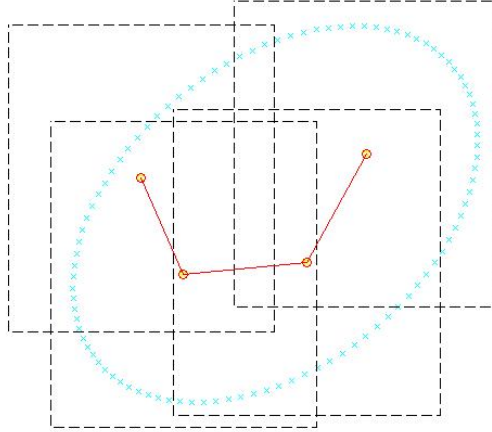


Figura 4.2: Resultado usando MILP, optimizando la distancia y un sensor con huella rectangular

La optimización puede utilizar las restricciones de la trayectoria que debe seguir el UAV [RAdlCP06]. En la figura 4.3 se observa la optimización para 4 puntos de observación y dos UAVs teniendo en cuenta las restricciones dinámicas de los vehículos.

Para el caso de más de 5 puntos de observación el computo es intratable pero podemos obtener soluciones que no optimizan en distancia como se presenta en

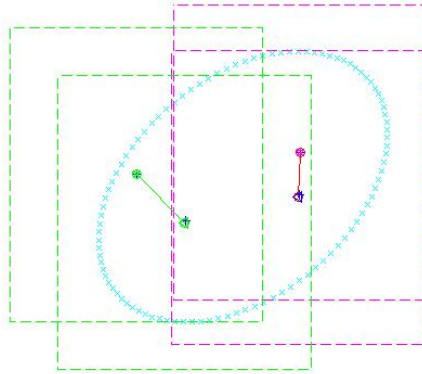


Figura 4.3: Resultado usando MILP cuatro puntos de observación y un sensor con huella rectangular

la figura 4.4, donde se puede observar que se cumplen las restricciones dinámicas pero el camino no es óptimo en distancia.

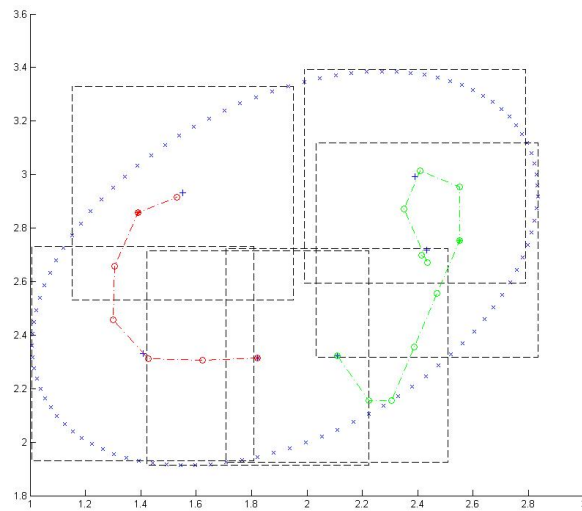


Figura 4.4: Resultado de la trayectoria para dos UAVS con 6 puntos de observación y un sensor con huella rectangular

Como conclusión es factible usar esta técnica para pocos puntos de observación. La minimización del número de puntos de observación es computacional-

mente alto y en la práctica exponencial con el número de puntos. Para poder utilizar técnicas de optimización lineal, hay que proporcionar un número fijo de lugares de observación. Usando un límite superior para el número de puntos, que nos proporcionará una solución, en un pentium CORE 2 DUO con más de 5 puntos de observación para una superficie arbitraria el tiempo de cálculo para la solución sobrepasa los 100 segundos. Dado que estamos buscando una estrategia que nos permita tomar decisiones en tiempo real, vamos a buscar una solución subóptima por aproximación. El límite inferior de la solución será recorrer todo el contorno de la superficie, es decir que si el rango del sensor es muy pequeño estaremos realizando el seguimiento del contorno.

### 4.3. Smart Shape Tracking

La solución propuesta SST, es una solución geométrica que explota las propiedades matemáticas de una superficie ajustada al contorno que necesitamos identificar y monitorizar. El eje central de esta estrategia es la búsqueda de los puntos de observación. Así esta técnica describe como encontrar esos puntos usando una elipse como aproximación a la superficie. Para la definición del problema asumiremos que toda la información acerca del contorno es conocida. Aparte del algoritmo SST presentado a continuación, en la figura 4.5 se representa la estrategia como una caja negra donde la entrada es una superficie geométrica ajustada al contorno de la superficie que se quiere identificar y la salida está compuesta por el número de puntos de observación y la posición en el espacio de los mismos.

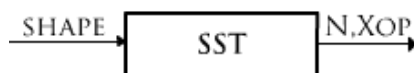


Figura 4.5: La entrada del módulo es la forma geométrica ajustada al contorno y la salida es el número de puntos de observación y su posición

Existe mucha documentación sobre shape fitting. Recientemente en visión por computador está siendo utilizada para el reconocimiento de caras. Pero también se está usando en geometría computacional [HPM03]. Una vez que tenemos la forma aproximada, podemos encontrar un mínimo número de puntos de observación que nos dará una solución subóptima.

La solución que se presenta es subóptima ya que se utiliza un contorno aproximado y no el real. La comparación con la solución óptima no será tratada en este documento y dependerá de el objetivo final de la tarea. La estrategia está pensada para su ejecución en tiempo real, y para el uso en enjambres de agentes como un problema de asignación (allocation problem).

### 4.3.1. Algoritmo SST

Asumimos que la superficie es conocida de antemano, por medio de una identificación (un recorrido horario, un algoritmo de ajuste [SBM08], etc). La comunicación está garantizada en cada paso de comunicación, ya que dado el polinomio de mínima distancia calculado implica que los puntos de observación están lo más cerca posible para la completa identificación. Si no existe comunicación hay que establecer puntos de encuentro (Rendez-vous). En el algoritmo 4.3.1 queda detallado en alto nivel el SST, en el caso de que la superficie ajustada sea una elipse. Ésta ha sido utilizada ya que es una superficie sencilla pero que permite el almacenamiento de mucha información y además proporciona propiedades muy interesantes para la partición de su contorno. La elipse se ajusta muy bien a superficies del tipo de manchas de petróleo y fuegos. La tarea de identificación en el seguimiento termina cuando todos los puntos de observación han sido visitados. Para el almacenamiento de puntos de interés de la imagen obtenida se utiliza una cola con factor de olvido, donde los puntos que no han sido detectados durante un tiempo, dejan de existir. Dado que suponemos dinámicas lentas, se utiliza un margen de error para el cálculo de los puntos de observación de tal manera que cuando el agente llegue al punto de observación el contorno seguirá siendo capturado. La superficie ajustada se guarda en una estructura ( $s$ ). El estado de los agentes se define como el vector  $(u(t)^1, \dots, u(t)^n)$ . El conjunto de puntos de observación es  $X_{op}$  y el número de puntos de observación es  $NX_{op}$ . Las tareas a realizar por los agentes son un conjunto de destinos para cada uno que se define como  $TA^i$ . La función de partición mencionada en el algoritmo se profundiza en la sección 4.3.2.

### 4.3.2. Particiones de la superficie usando una elipse

Sean  $p_k \in \partial B$  los puntos frontera de la superficie,  $X_{op}$  definido como un conjunto de puntos de observación, que tienen la propiedad de que desde allí se puede capturar todo el contorno del objeto. Los puntos de observación tienen que cumplir la condición definida en 4.3 y 4.4

Puede ser también visto como la mínima trayectoria definida por puntos en el espacio para identificar la superficie. Si  $field\_of\_view(s(o))$  es el campo de visión del sensor  $s$  con origen en  $o$ , se cumple

$$\bigcup p_k \in field\_of\_view(s(X_{op})) = \partial B$$

Para encontrar estos puntos de observación vamos a utilizar una elipse como forma que se ajusta a nuestro objeto. Sabemos que una bounding-box (mín  $p_k \in \partial B$ , máx  $p_k \in \partial B$ ) como forma es muy simple de ajustar, pero proporciona poca información. Sin embargo la elipse posee mucha más información y se puede ajustar mucho mejor a formas como una mancha de petróleo en el mar. Las

**algorithm 2** Algoritmo SST

---

```

 $p = points\_extraction(I)$ 
mientras no fin
     $e = ellipse\_fitting(p)$ 
     $thr = error\_estimation(u, e)$ 
    Puntos de observacion
    si  $2r < e.longaxis$ 
         $X_{op} = center(e)$ 
    sino si  $2r < e.shortaxis$ 
         $X_{op} = partitions\_case2(e, r)$ 
    sino
         $NX_{op} = upperbound(e, r)$ 
         $X_{op} = partitions\_case3(e, r)$ 
    fsi
    Asignacion de los puntos de observacion a los agentes
     $TA = allocate(X_{op})$ 
    mientras no identificado
        si en punto de observación
             $p = points\_extraction(I)$ 
    fmientras
fmientras

```

---

propiedades matemáticas de la elipse son explotadas para conseguir una buena solución subóptima para el tracking en tiempo real de los puntos frontera de una superficie geográfica. El perímetro es dividido en particiones. Este número de particiones depende de la longitud del perímetro y del tamaño del sensor. Como precondition imponemos que siempre se puede ajustar una elipse y que el sensor es una cámara, que captura imágenes cuadradas.

Tres casos tienen que ser diferenciados, dependiendo del radio del sensor para obtener una mejor solución: no sobredimensionado, sobredimensionado en un eje, sobredimensionado en dos ejes(caso general). Definimos la elipse  $E$  que se ajusta al contorno y que posee estos parámetros:

- $\phi$  - Orientación en radianes de la elipse (tilt)
- $longaxis$  - Tamaño del eje mayor de la elipse
- $shortaxis$  - Tamaño del eje menor de la elipse

**No sobredimensionado**

Sea  $r$  el radio de acción del sensor  $2r > longaxis$ . Implica que todo el contorno está cubierto por el sensor. La solución será un único punto, el centro de la elipse.

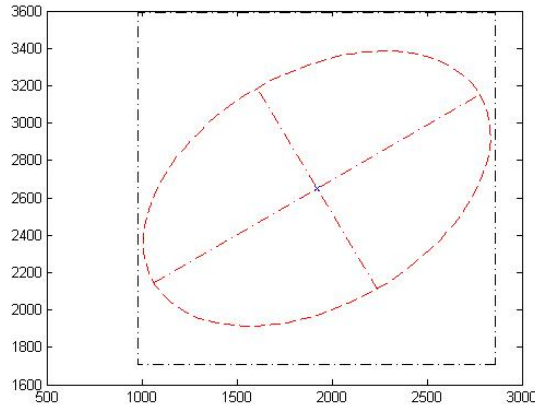


Figura 4.6: SST, Solución cuando el sensor es más grande que la superficie geográfica

### Sobredimensionado en una dimensión

Sea  $r$  el radio de acción del sensor  $2r > \text{shortaxis}$ .

La solución viene dada por encontrar los puntos en el eje largo de la elipse que minimice la distancia. Dependiendo del signo de  $\phi$  la dimensión sobredimensionada es  $x$  o  $y$ . Los puntos de observación pueden ser encontrados siguiendo el método siguiente:

Sea  $mp$  el punto mínimo de la elipse en el eje largo  $mp = \min p_k$  donde  $p_k \in E$ , con una matriz de rotación  $R$

$$R = \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix}$$

Encontramos los puntos que cumplen esta restricción:  $pp = R \cdot mp \mid \|mp - pp\| = r$ , donde  $r$  es el radio de la huella del sensor. La restricción busca por el punto no rotado a distancia  $r$  desde el mínimo punto del contorno en la dirección del eje. Finalmente la proyección rotada  $\pi$  de  $pp$  en el eje largo de la elipse no rotada proporcionará el punto inicial.

$$pstart = R\pi$$

Si el doble de la distancia calculada desde el punto extremo en el eje largo de la elipse al punto de comienzo es mayor que el eje largo, implica que la elipse está contenida en la diagonal del sensor cuadrado, así la solución es la misma que en el primer caso.

En otro caso el eje largo es subdividido en particiones de tamaño  $r$  comenzando en el punto  $pstart$  y terminando en  $pend$  (que se encuentra de manera simétrica que  $pstart$ ).

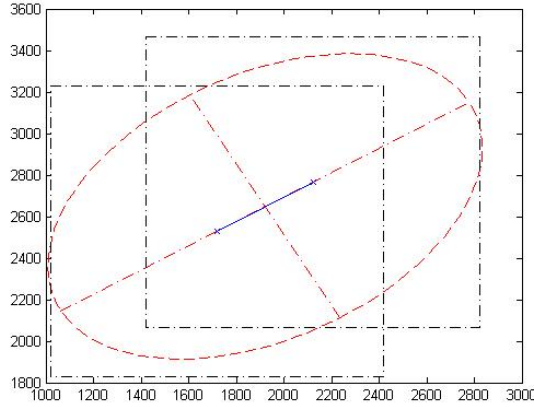


Figura 4.7: SST, Solución cuando el sensor es más grande en una dimensión que la superficie geográfica

#### Caso general, sobredimensionado en dos dimensiones

El contorno es dividido en particiones utilizando un límite superior para el número de puntos de observación.

Sea  $Xop$  los puntos de observación;  $r$  el radio del sensor;  $P$  el perímetro de la elipse ajustada;  $bbox$  la caja contenedora del objeto;  $A_{ellipse}$  el área de la elipse;  $A_{sensor}$  el área del sensor. El límite superior para una solución válida es definido como se muestra en Eq. 4.5.

$$\min \{p/r, P(bbox)/r, A_{ellipse}/A_{sensor}\}$$

La solución será un conjunto de puntos y de ángulos de rotación del sensor en el caso de que el sensor no sea circular. El ángulo es definido por el arco entre las dos particiones y depende de su curvatura. Es la línea que une el punto medio de las dos particiones y el punto solución. Así el lado superior del rectángulo que define la huella del sensor será una línea tangente a la elipse.

Sea  $p_k$  los puntos de partición,  $C_e$  el centro de la elipse rotada,  $d_k = \|p_k - C_e\|$  la distancia entre el punto de partición y el centro de la elipse, y  $\vec{v}_k$  el vector normalizado del segmento  $\overline{p_k C_e}$ . la solución es:

$$Xop_k = C_e + d_k \cdot \vec{v}_k \quad (4.6)$$

#### 4.4. Estudio de la altura y la huella del sensor

Está demostrado que la proporción de radio del sensor sobre el perímetro de la superficie ajustada define el número de puntos de observación y por lo tanto la trayectoria.



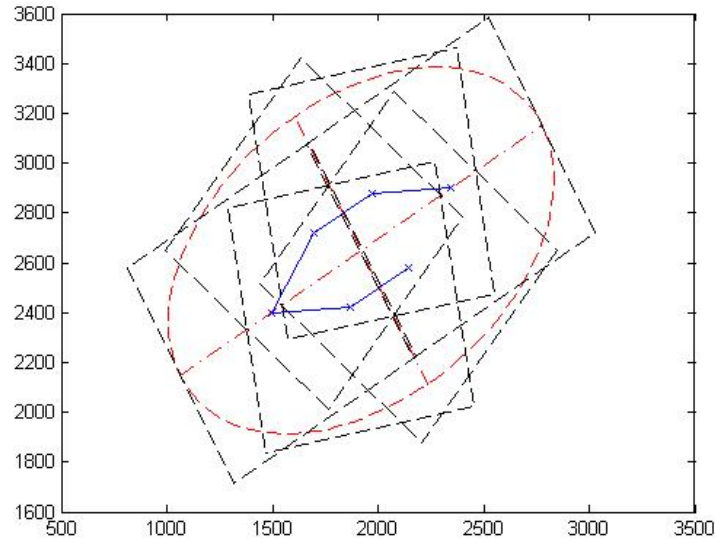


Figura 4.8: SST, Solución general haciendo una partición del perímetro

Sabemos que el sensor cubrirá el contorno si está entre el 15 % y el 20 %. Con la elipse como la superficie ajustada y un cuadrado como huella del sensor, tenemos que la diagonal de la huella y el ángulo de rotación de la elipse modifican el número de puntos de observación.

Teniendo un campo de vista fijo, el radio sólo depende de la componente  $Z$  del sensor. Es decir la altura. La cámara está definida por el índice de muestreo, el número de píxeles, ángulo del campo de vista y la distancia focal.

Existen dos casos especiales: cuando el campo de vista es muy pequeño será lo mismo que realizar un seguimiento del contorno; en el caso opuesto el campo de visión es suficiente para cubrir todo el contorno.

El límite para modificar la altura de la cámara en UAV es: el ruido producido por estar más lejos del objetivo, la dinámica del UAV y los límites del entorno. La figura 4.9 muestra como la estrategia funciona en un contorno que se mueve. La altura del UAV es suavemente incrementada, pero la dinámica del contorno es rápida y el número de puntos de observación aumenta. Notar que el contorno es ajustado por una elipse y que el contorno no es un círculo. El contorno se mueve lateralmente y longitudinalmente, además de expandirse de acuerdo con las corrientes del mar y las componentes de viento.

El índice de perímetro - radio en 4.10, demuestra empíricamente que en 0.2 en el eje  $y$  (es decir la proporción perímetro - radio), el coste de la trayectoria comienza a crecer. El número de puntos de observación también crece. Así en cuanto se alcanza el 20 % de proporción del perímetro, se consigue mejor rendimiento.

Se puede observar que en el instante (time=40) que la trayectoria de seguir el

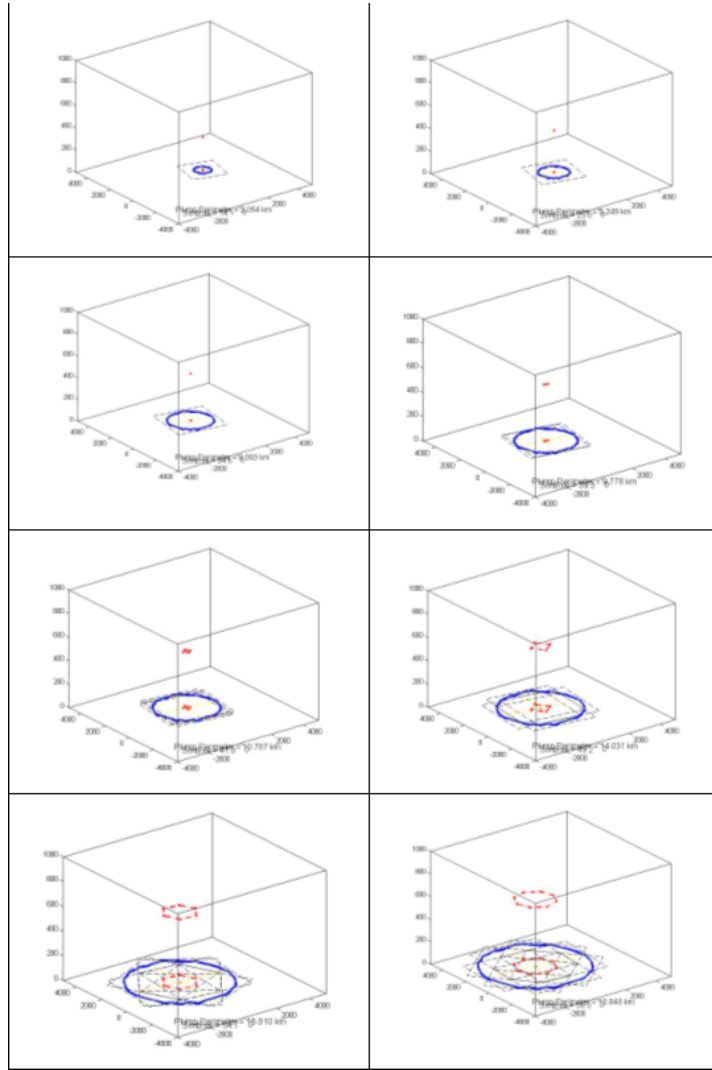


Figura 4.9: Capturas de la simulación usando SST para hayar los puntos de observación con una superficie que varia con el tiempo y un sensor que puede incrementar su altura

contorno entero es de 10000 metros; mientras tanto con esta estrategia los valores son cercanos a 0 metros, salvando 10 km de recursos. Si se tiene que realizar un seguimiento continuo 10 km son salvados en cada pasada.

#### 4.5. Política de asignación para múltiples UAVs

El seguimiento realizando esta estrategia y añadiendo un manejador de asignación de tareas (todos los agentes tienen que estar trabajando si existe alguna tarea que realizar) será simulada para probar su rendimiento. Otras políticas de asignación permite tener algunos de los agentes cargando las baterías mientras

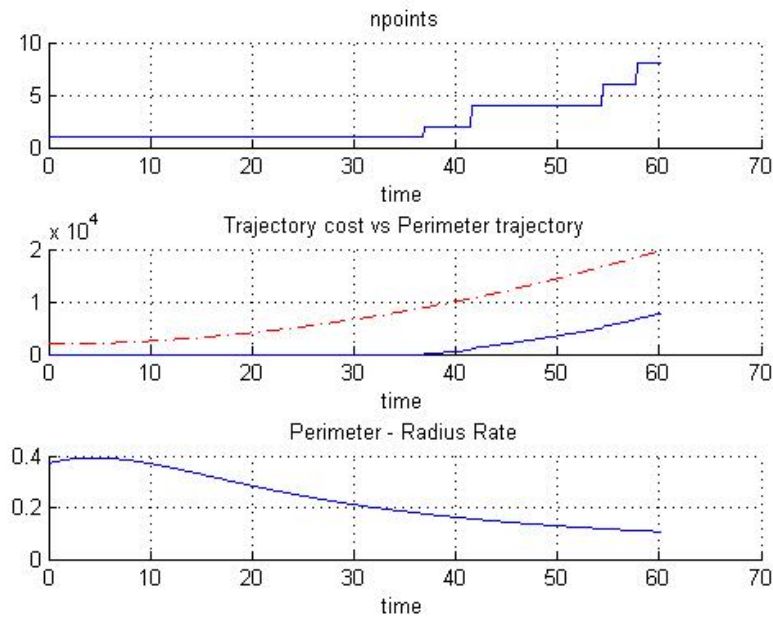


Figura 4.10: Capturas de la simulación usando SST para hallar los puntos de observación con una superficie que varía con el tiempo y un sensor que puede incrementar su altura

que el contorno está siendo identificado por el resto.

Dos escenarios son propuestos. El primero, extraído de [SBM08], es un contorno no convexo. El segundo es una modelada mancha de petróleo que tiene en cuenta factores de mar y viento. El primer escenario es representativo para el seguimiento de contornos no convexos y el segundo se usará para ver el rendimiento de la estrategia ante contornos de dinámicas más rápidas y que se desplazan en el plano  $XY$ . Asumimos que los agentes conocen el contorno en el inicio. El algoritmo es comparado con un idílico controlador de seguimiento de contorno.

#### 4.5.1. Definiciones

El concepto de **tareas** es introducido para calcular el rendimiento del algoritmo. Una tarea contempla una completa identificación del contorno. Por ejemplo, en el seguimiento de contorno la tarea será completada cuando se realice cuando el primer robot llegue a la posición inicial del segundo robot. Así si sólo hay un agente la tarea será una vuelta en el sentido horario. Utilizando la estrategia SST, una tarea será completada cuando todos los miembros del enjambre se encuentran en estado idle, que significa que el contorno ha sido identificado completamente.

Las **reglas de asignación de tareas** se refieren a cómo cada conjunto de puntos de observación es asignado a cada entidad en el enjambre. En este caso los

puntos de observación serán asignados en orden a cada entidad. Si hay menos puntos de observación que agentes, los agentes sin tarea irán a un punto específico con comunicación y esperarán nuevas tareas. Si el número de puntos de observación es mayor que los agentes, un subconjunto de puntos de observación serán asignados en orden a cada entidad.

La **distancia acumulada del enjambre** es la distancia acumulada para todos los agentes tratado como una única entidad. Esto será utilizado para estudiar el consumo de recursos (Batería, combustible, etc).

#### 4.5.2. Contorno de variación lenta en el tiempo

En este escenario de simulación se usa un enjambre donde cada agente está descrito por las mismas dinámicas. El tiempo de simulación es 200s, el paso de tiempo es 0,1s, el radio del sensor es 35. El contorno no convexo que varia con el tiempos  $\partial B(t)$  está descrito por

$$\gamma(\phi, t) = \left( 2\left(1 - \frac{t}{t_f}\right) + (2 + \cos(10\pi\phi) + 0,5 \sin(4\pi\phi))\frac{t}{t_f} \right) \begin{pmatrix} \cos(2\pi\phi) \\ \sin(2\pi\phi) \end{pmatrix} \quad (4.7)$$

Será utilizado como el objeto a seguir.

La figura 4.11 muestra el algoritmo ejecutándose en diferentes instantes incluyendo el estado inicial y final. El número de agentes es 3 y están representados por triángulos, blanco cuando están en estado idle y rojo cuando tienen una tarea que realizar. Se muestra la configuración del enjambre en los instantes  $t = 0$ ,  $t = 50$ ,  $t = 100$  y  $t = 200$ . En la configuración inicial los agentes se encuentran en el contorno e intentarán posicionarse en el centro, formando un polígono mínimo. Este caso está elegido porque siempre hay un agente en movimiento.

La figura 4.12 compara el rendimiento entre la estrategia SST y un sigue contornos ideal. La barra azul representa el seguimiento de la forma ajustada presentada en este capítulo. La distancia acumulada del enjambre es obviamente mayor en el otro controlador, excepto en las dos primeras simulaciones, con uno y dos agentes respectivamente. Se necesita mirar los dos gráficos de barras para explicar el fenómeno. El número de tareas de identificación terminadas con un agente utilizando SST es más alto, que significa que para el mismo índice de tareas por segundo la distancia acumulada, la barra azul será siempre menor que la roja. Las tareas por segundo es asumiendo que la acción realizada por cada agente para la detección es menor que el tiempo de cada paso. En los casos donde esto necesita más tiempo, como utilizando algoritmos complejos de visión por computador, las tareas por segundo serán modificadas por el tiempo de procesamiento, pero las proporciones entre los rendimientos seguirán siendo casi los mismos.

Con un enjambre de 10 agentes 1508 metros son acumulados con la estrategia propuesta, mientras que con el seguimiento del contorno se acumulan 15902

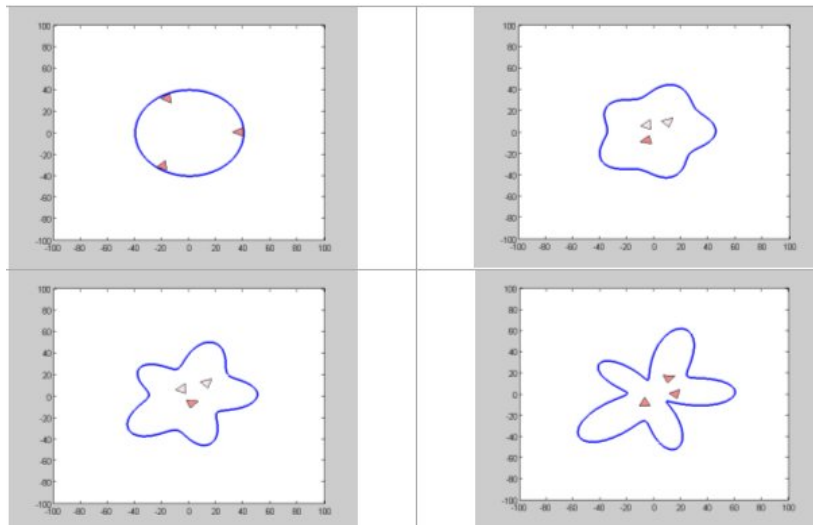


Figura 4.11: La figura muestra el algoritmo ejecutándose en diferentes instantes incluyendo el estado inicial y final. El número de agentes es 3 y están representados por triángulos, blanco cuando están en estado idle y rojo cuando tienen una tarea que realizar.

metros. Un orden menos o 10 veces menos, así que los recursos que se salvan son bastante altos.

### 4.5.3. Mancha de petróleo

Para este experimento se asume que la dinámica del objeto a seguir son más lentas, comparadas con la dinámica del agente. En [SS07, WSZ05] la dinámica de las manchas de petróleo es presentada. Utilizando estos conceptos un contorno es generado y se mueve conforme a componentes de viento y mar. En la simulación se tienen en cuenta dos fases de la dinámica del objeto. Los primeros 100s las dinámicas son rápidas y en los siguientes 100s las dinámicas son lentas.

En la figura 4.13 y la figura 4.14 un comportamiento diferente es mostrado. Como el contorno es mayor que en el otro caso, incluso con 10 agentes, tienen que moverse a lo largo de todo el contorno para seguir el borde. También se tienen que desplazar en la misma dirección que el objeto, lo que implica que siempre se están moviendo. El rendimiento en la distancia acumulada y las tareas por segundo siguen siendo mejores que en el caso de seguir todo en contorno. De hecho la distancia acumulada es muy dependiente del viaje del objeto. Si las dinámicas son más lentas el rendimiento del algoritmo serán más altas.

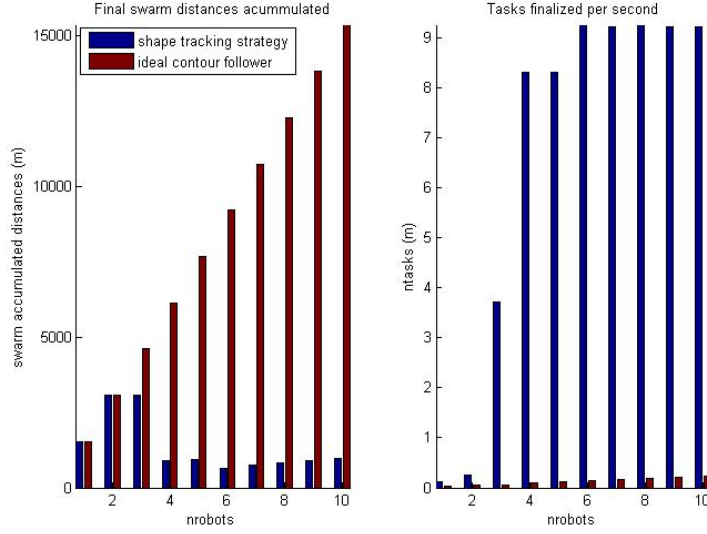


Figura 4.12: Estas dos figuras tienen que ser analizadas al mismo tiempo. En azul es el SST y en rojo es el seguimiento de contorno ideal. En la izquierda se muestra la distancia acumulada del enjambre, la suma de todo el camino realizado de los agentes. En la derecha se muestra el número de tareas de identificación realizadas por segundo. Cada grupo de barras representan una simulación diferente de 200 segundos con diferente número de robots con sensores.

#### 4.6. SST con información local

En esta sección se introducen algunas ideas sobre como se puede adaptar esta estrategia cuando no todos los UxVs tienen toda la información sobre el entorno.

Utilizar una red de comunicaciones con sensores ha sido estudiado recientemente [OFL04, GK05] y aplicado a optimización [RN04]. Aquí se presenta la adaptación de intercambiar la información porque cada agente sólo posee información local.

Sea  $p_k \in \partial B$  los puntos que definen el contorno, la caja contenedora (bounding box) puede ser representada por estas dos funciones:

$$Mn = \min(p_k)$$

$$Mx = \max(p_k)$$

En vez de intercambiar toda la información y calcular el mínimo en cada nodo, se puede calcular el mínimo local de cada nodo y propagarlo.

Sea  $Mn_n$  el mínimo local de cada nodo y  $Mx_n$  el máximo local

$$Mn = \min_1^n(Mn_n)$$

$$Mx = \max_1^n(Mx_n)$$

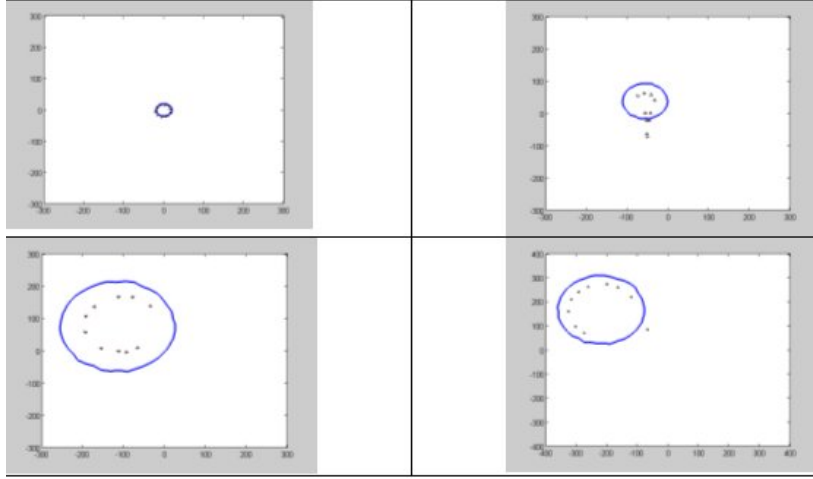


Figura 4.13: Esta figura muestra el enjambre en los instantes  $t=0$ ,  $t=50$ ,  $t=100$  and  $t=200$ , incluyendo los estados inicial y final. El número de robots es 10 y están representados por triángulos, blancos cuando están en estado idle y rojos cuando están realizando alguna tarea.

Cada agente computa su función local y en cada paso de comunicación la red calculará la función global. Así al final de la etapa de comunicación cada agente con sensor tendrá la función global como se muestra en la figura 4.15. Con la función global en cada nodo, se puede calcular el SST para la forma representada por la función. La política de asignación de los puntos de observación consiste en nombrar cada nodo y asignar en orden el número de puntos de observación.

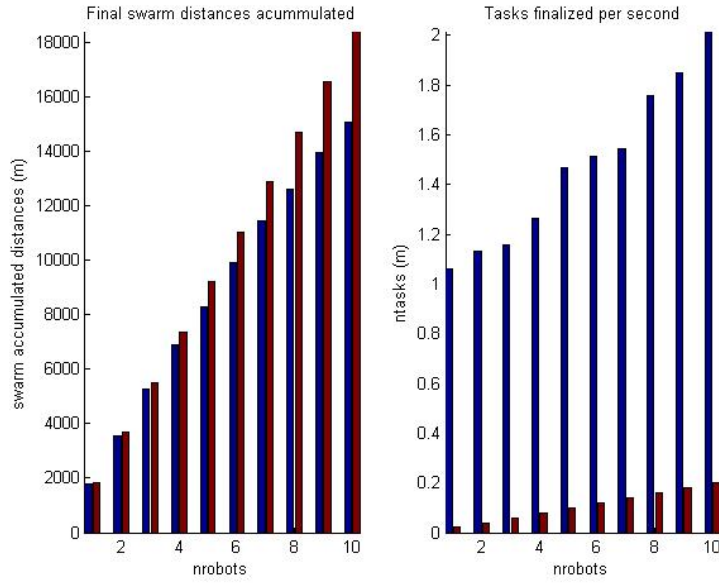


Figura 4.14: En azul se representa la estrategia SST y en rojo el seguimiento de contorno ideal. En la izquierda se muestra la distancia acumulada del enjambre, la suma de todo el camino realizado de los agentes. En la derecha se muestra el número de tareas de identificación realizadas por segundo. Cada grupo de barras representan una simulación diferente de 200 segundos con diferente número de robots con sensores.

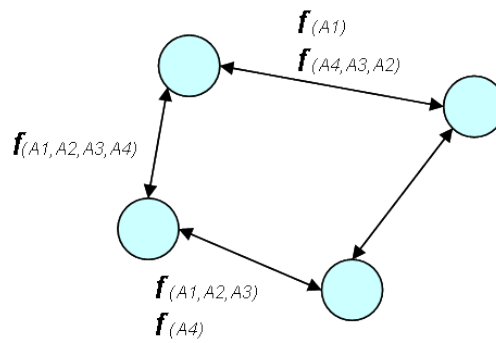


Figura 4.15: En el instante  $t$  de la etapa de comunicación la función es propagada.



## Capítulo 5

---

# Conclusiones

---

El resultado del trabajo realizado durante una estancia de 4 meses en el MIT (EEUU) queda plasmado en este documento. El proyecto CICYT (DPI2006-15661-C02-01) que ha permitido este trabajo de investigación titulado "Plataforma de planificación, simulación y control para colaboración de múltiples vehículos autónomos marinos y aéreos", es el inicio de un proyecto de mayor escala. Por ello destacaremos dos apartados importantes: Las aportaciones que se han llevado a cabo en el campo de la investigación y el trabajo futuro que se realizará tomando como base este documento.

### 5.1. Aportaciones

Así como en la introducción se comentó la subdivisión de la identificación y el seguimiento de superficies geográficas con visión artificial en diferentes subproblemas, me referiré del mismo modo en esta sección. Esta subdivisión ha sido realizada conforme a un minucioso estudio de la literatura acerca de problemas similares.

- Dentro de la tarea de búsqueda hay que reseñar que se introduce un novedoso algoritmo de contención de objetivos dinámicos bajo incertidumbre para multi-UAV. Aparte conforme a un estudio y revisión de técnicas de búsqueda con incertidumbre, se ha creado una estrategia de búsqueda híbrida basada en la definición de un controlador PD donde la parte proporcional es la búsqueda de la máxima probabilidad y la parte diferencial se refiere a la estrategia de contención. Este nuevo algoritmo se ha desarrollado para trabajar en tiempo real mediante una optimización de horizonte infinito con optimalidad local temporal. Finalmente se propone un seguimiento activo

donde el UAV influirá directamente sobre el entorno para la monitorización de manchas de petróleo o fuego.

- En la tarea de identificación se ha realizado un estudio de técnicas de identificación de contorno y su posible aplicación o especialización para las superficies geográficas utilizando visión artificial, diseñando una arquitectura con un generador de trayectorias de alto nivel que permite flexibilidad a la hora de tomar decisiones y la total transparencia al programador a la hora de programar un UAV. Esto se ha probado mediante el desarrollo de un generador de trayectorias para identificar el contorno de una superficie sobredimensionada en sentido horario. Aparte se introduce la posibilidad de ajustar una superficie geométrica para su posterior monitorización.
- Bajo la tarea de seguimiento y monitorización se ha diseñado e implementado una nueva estrategia para la monitorización de superficies dinámicas sobredimensionadas con multiagentes, explotando el campo de acción del sensor, para el seguimiento prolongado. El problema ha sido definido en MILP para el posterior desarrollo de un algoritmo subóptimo basado en el número de puntos de observación y la asignación ordenada sobre los agentes.
- Se ha implementado una plataforma para probar los algoritmos mencionados para la identificación y seguimiento de superficies geográficas con cierto comportamiento dinámico, utilizando visión artificial. La implementación esta realizada en Matalab Simulink.

## 5.2. Trabajo Futuro

Las numerosas vías de investigación que se abren debido a este documento conforman un buen comienzo para el proyecto. Así se pueden enumerar como posibles ideas para continuar trabajando, las siguientes.

- Adaptación de la estrategia híbrida de búsqueda para la identificación de manchas de petróleo y monitorización de fuegos.
- Desarrollo de un algoritmo óptimo global para entornos dinámicos con incertidumbre.
- Investigar técnicas de coordinación con enjambres de agentes
- Desarrollo de técnicas de identificación y seguimiento robustas.
- Inclusión de métodos de aprendizaje.
- Implementación de los algoritmos para uso real con UAVs en el exterior
- Inclusión de métodos o estrategias para esquivar obstáculos inesperados

- Inclusión de obstáculos dinámicos con incertidumbre.
- Desarrollo de una plataforma sólida para probar algoritmos de coordinación entre UAVs



---

# Bibliografía

---

- [Bea08] Randal Beard. *Quadrotor Dynamics and Control Rev 0.1*, 2008. [cited at p. 68, 69]
- [BH06] L.F. Bertuccelli and J.P. How. Search for dynamic targets with uncertain probability maps. In *Proc. American Control Conference*, pages 6 pp.–, 2006. [cited at p. 9, 10]
- [CBM<sup>+</sup>05] D.W. Casbeer, R.W. Beard, T.W. McLain, Sai-Ming Li, and R.K. Mehra. Forest fire monitoring with multiple small uavs. In *Proc. American Control Conference the 2005*, pages 3530–3535 vol. 5, 2005. [cited at p. 7, 41]
- [CF05] J. Clark and R. Fierro. Cooperative hybrid control of robotic sensors for perimeter detection and tracking. In *Proc. American Control Conference the 2005*, pages 3500–3505 vol. 5, 2005. [cited at p. 5, 41]
- [Fre04] Eric W. Frew. Flight demonstrations of self-directed collaborative navigation of small unmanned aircraft. 2004. [cited at p. 5]
- [GK05] A. Giridhar and P.R. Kumar. Computing and communicating functions over sensor networks. 23(4):755–764, 2005. [cited at p. 56]
- [HPM03] Sarel Har-Peled and Soham Mazumdar. Fast algorithms for computing the smallest k-enclosing disc. In *Lecture notes in computer science*, 2003. [cited at p. 46]
- [McL08] James McLurkin. *Analysis and Implementation of Distributed Algorithms for Multi-Robot Systems*. PhD thesis, MIT, 2008. [cited at p. 42]
- [OFL04] P. Ogren, E. Fiorelli, and N.E. Leonard. Cooperative control of mobile sensor networks:adaptive gradient climbing in a distributed environment. 49(8):1292–1302, 2004. [cited at p. 56]
- [PROL<sup>+</sup>08] G. Pajares, J.J. Ruz Ortiz, P. Lanillos, M. Guijarro, J.J. De la Cruz, and M. Santos. Generacion de trayectorias y toma de decisiones para uav's. *RIAI*, 5:83–92, 2008. [cited at p. 4, 43]
- [RADlCP06] J.J. Ruz, O. Arevalo, J.M. de la Cruz, and G. Pajares. Using milp for uavs trajectory optimization under radar detection risk. In *Proc. IEEE*

- Conference on Emerging Technologies and Factory Automation ETFA '06*, pages 957–960, 2006. [cited at p. 4, 44]
- [RAPdlC07] J.J. Ruz, O. Arevalo, G. Pajares, and J.M. de la Cruz. Decision making among alternative routes for uavs in dynamic environments. In *Proc. ETFA Emerging Technologies & Factory Automation IEEE Conference on*, pages 997–1004, 2007. [cited at p. 4]
- [RN04] M. Rabbat and R. Nowak. Distributed optimization in sensor networks. In *Proc. Third International Symposium on Information Processing in Sensor Networks IPSN 2004*, pages 20–27, 2004. [cited at p. 56]
- [SB95] S.M. Smith and J.M. Brady. Susan - a new approach to low level image processing. Technical Report TR95SMS1c, Chertsey, Surrey, UK, 1995. [cited at p. 37]
- [SBM08] S. Susca, F. Bullo, and S. Martinez. Monitoring environmental boundaries with a robotic sensor network. 16(2):288–296, 2008. [cited at p. 33, 41, 47, 53]
- [SS07] P. Sebastiao and Guedes Soares. Uncertainty in predictions of oil spill trajectories in open sea. *Ocean engineering*, 34:576–584, 2007. [cited at p. 55]
- [WSZ05] Wang, Shena, and Zhengb. Two-dimensional numerical simulation for transport and fate of oil spills in seas. *Ocean engineering*, 32:1556–1571, 2005. [cited at p. 9, 55]
- [YMP02] Yanli Yang, Ali Minai, and Marios Polycarpou. Decentralized cooperative search in uav's using opportunistic learning. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2002. [cited at p. 9]
- [YMP04] Yanli Yang, A.A. Minai, and M.M. Polycarpou. Decentralized cooperative search by networked uavs in an uncertain environment. In *Proc. American Control Conference the 2004*, volume 6, pages 5558–5563, 30 June–2 July 2004. [cited at p. 9, 10]

## Apéndice A

---

# Seguimiento activo, aplicación para superficies sobredimensionadas

---

El seguimiento activo se define como una estrategia de contención para monitorizar superficies geográficas. Es activo ya que estamos modificando el entorno conforme realizamos detecciones. En todo momento estamos hablando de búsqueda de objetos dinámicos, pero si el objetivo en vez de moverse sin modificar su tamaño, se expande, la estrategia de contención puede ser utilizada para su monitorización. Numerosas aplicaciones como identificación de manchas de petróleo y monitorización de fuegos, pueden ser tratadas con estrategias de contención de la probabilidad. Es más, usando el gradiente, podemos averiguar que zonas tienen un mayor incremento y por tanto hay que visitar primero. Otra forma de interpretarlo es teniendo en cuenta la escala. La detección de una superficie sobredimensionada depende de la discretización del entorno. Así se utilizará una búsqueda del objetivo con un tamaño de celda que contenga la superficie (**macro-escala**, nos interesa la posición de la superficie y no su forma), y una vez encontrada se procederá a reducir la escala para su correcta identificación y seguimiento (**micro-escala**, nos interesa la forma detallada del objeto).

### A.1. Aplicación sobre manchas de petróleo

Un ejemplo muy atractivo del seguimiento activo, es decir, que estamos interaccionando con el entorno de manera activa, es el uso de toallas de papel para la absorción del petróleo. Francesco Stellaci, un experto en nanotecnología del departamento de materiales del MIT, asegura que han inventado un papel for-

mado por nanohilos que pueden absorber selectivamente hasta 20 veces su peso en petróleo, del agua. Utilizando este tipo de papel, se podría efectuar un algoritmo de contención donde se distribuyera este tipo de papel absorbente, incidiendo activamente en la erradicación del problema.



## Apéndice B

# Implementación del sistema

La arquitectura diseñada para la búsqueda e identificación de superficies terrestres forma una plataforma para la implementación de las estrategias desarrolladas en los capítulos anteriores. El diseño se basa en modelos de entrada / salida con cuatro bloques bien diferenciados (figura B.1):

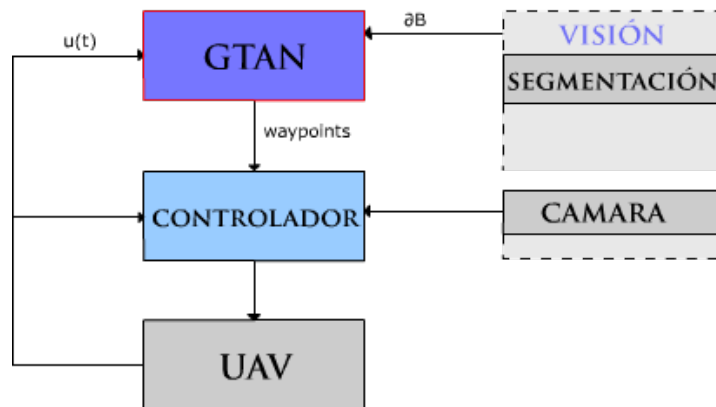


Figura B.1: Diseño de la arquitectura del sistema, modelos e interconexiones.

1. Sistema de Visión: Permite la captura de imágenes y el filtrado de las mismas. Se puede dividir en dos partes:
  - Adquisición de imágenes: Consta de una cámara que es el sensor principal del UAV para proporcionar información. El retardo de la captura de las imágenes es modelado.
  - Procesamiento de imágenes. Dada una imagen capturada en el proceso de adquisición, se somete a varias etapas de filtros y se extrae la información necesaria.

2. Generador de trayectorias de alto nivel: Conformar el cerebro de la arquitectura, ya que es donde se toman las decisiones. En él se calcularán las trayectorias que debe seguir el UAV para realizar la tarea asignada.
3. Quatrimotor: es el modelo del UAV que vamos a utilizar extraído de [Bea08]. En este bloque residen tanto los sensores como los actuadores del UAV. Contiene el controlador de bajo nivel que garantiza la llegada a los puntos definidos por el generador de trayectorias de alto nivel.
4. Superficie: Define la información que poseemos sobre la superficie a identificar, su dinámica y sus propiedades.

### B.1. Etapa de filtrado

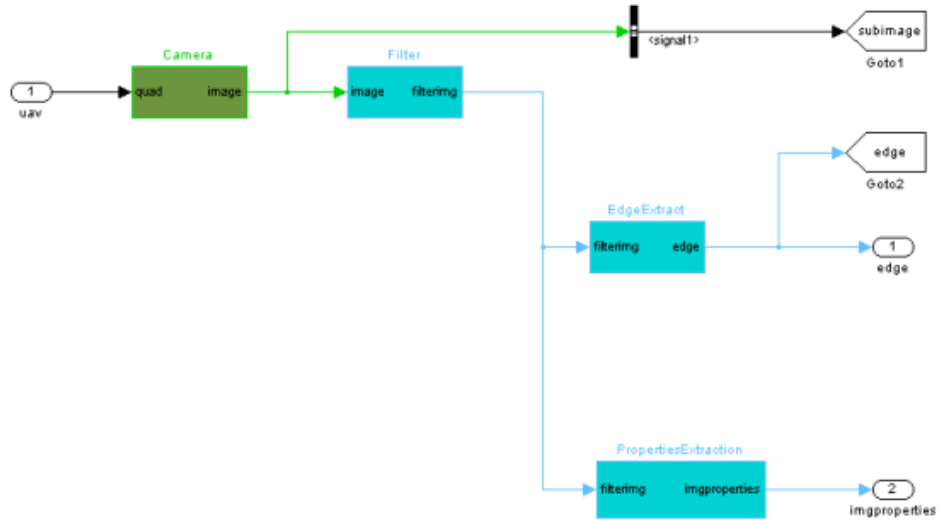


Figura B.2: Diagrama de matlab simulink de la etapa de filtrado

La etapa de filtrado será la encargada de segmentar la imagen y darnos la información que necesitamos. Dada una imagen como entrada, los pasos a seguir en el filtrado son los siguientes (el diagrama de cajas se puede ver en la figura B.2):

1. Conversión a sistema HSV para evitar que los cambios de luz afecten a la detección.
2. Realizar un estudio de la intensidad por medio de un histograma para crear un filtro robusto ante los cambios de luz.

3. Filtrado binario con  $\alpha^{sup}$  proporcionando la imagen binaria con los valores por debajo de  $\alpha^{sup}$
4. Filtrado binario con  $\alpha^{inf}$  proporcionando la imagen binaria con los valores por debajo de  $\alpha^{inf}$
5. Imagen binaria final calculada como:

$$I = \neg I_{bin}^{\alpha^{sup}} - \neg I_{bin}^{\alpha^{inf}}$$

6. Erosión y dilatación

$$I = I \oplus K_1 \oplus K_2 \circ K_3$$

7. Segmentación

- Extracción de puntos de interés o contorno  $\partial(I)$
- Extracción de propiedades: Centro de Masas ( $CM(I)$ ) y Orientación( $\phi(I)$ )

## B.2. Generador de trayectoria de alto nivel

El controlador de alto nivel se encarga de decidir los puntos en el espacio donde se tiene que dirigir el UAV, conforme a la información del sistema y los datos recogidos por los sensores. En este bloque se implementa la estrategia que se quiere utilizar. Como ejemplo en la figura B.3 la identificación con un ajuste de elipse y su posterior monitorización utilizando SST. En la imagen superior se muestra la región de búsqueda y los puntos de interés que se están extrayendo. El círculo rojo representa el siguiente waypoint que tiene que alcanzar el UAV. En la imagen inferior izquierda se muestra la representación 3D del UAV donde se puede observar la huella de la cámara. Además se observa la elipse ajustada a la superficie. En la imagen inferior derecha se muestra la imagen que está capturando la cámara situada en el UAV.

## B.3. Controlador del quadrimotor

El controlador de bajo nivel es la capa que nos permite transparencia respecto a los actuadores del UAV, es decir, es el que se encarga de asegurar que el UAV pasa por los puntos que hemos decidido en el controlador de alto nivel.

El controlador está basado en la documentación de [Bea08]. Es un controlador para un quadrimotor UAV y sería el controlador de bajo nivel, que recibirá como entrada los puntos de la trayectoria que debe seguir.

Se utilizará el estado estimado para el controlador, pero en vez de usar un filtro (Filtro extendido de Kalman) para estimarlo, el sistema de seguimiento VICON es utilizado para conocer el estado actual del UAV.

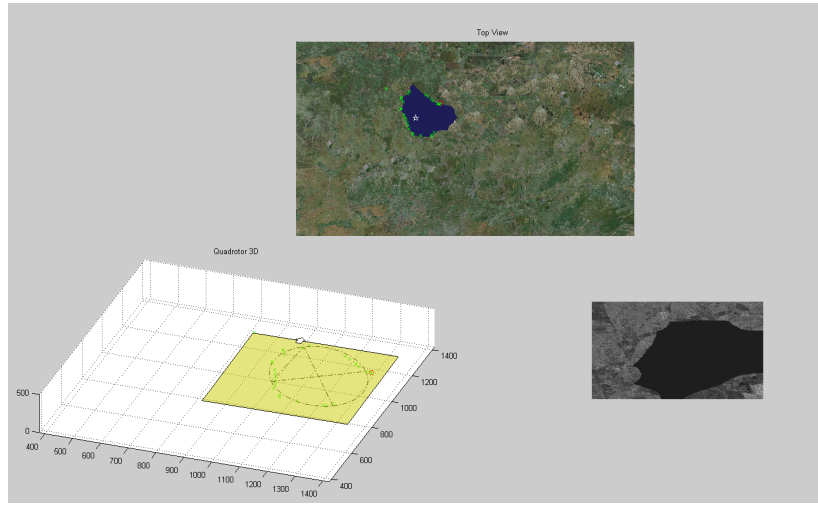


Figura B.3: Muestra un instante de tiempo en la simulación utilizando SST para la monitorización de la superficie

El estado es:

$$f(x, u) = \begin{pmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \\ \theta a_z \\ -\theta a_z \\ g + a_z \\ p \\ q \\ r \end{pmatrix} \quad (\text{B.1})$$

El diseño de control es una estrategia de PID para las componentes  $ux, uy$  y  $uz$ . A partir de estas componentes, se procederá a calcular la fuerza deseada  $F$ , el ángulo de cabeceo  $\phi^c$  y balanceado  $\theta^c$ .

### B.3.1. Control de la altitud por visión

Para el control de altitud se necesita derivar una expresión para  $uz$  para generar  $pz$  a la altitud deseada basada en el tamaño del objeto que se quiere identificar y la altitud máxima a la que se puede volar sin afectar a la detección. Se necesita medir el error de ruido de la cámara para calcular esa altura máxima a la que el UAV puede ir.

El radio debe ser  $r = \alpha \text{Perímetro}$ . El radio depende del ángulo del campo de visión y la componente  $Z$ .

Teniendo el plano de la imagen a la distancia focal  $f$ ,  $r_0$  es definida. Usando triángulos equivalentes tenemos:

$$\frac{-P_z}{r} = \frac{f}{r_0}$$

$$p_z = \frac{f\alpha\text{Perímetro}}{r_0}$$

Debido a los límites de la altura

$$p_z = \max\left(\frac{f\alpha\text{Perímetro}}{r_0}, \text{limite\_de\_ruido}\right)$$

Diferenciando podemos derivar el controlador PID.

### B.3.2. Control del balanceo

La ecuación de movimiento para el balanceo viene dada por:  $\ddot{\phi} = \frac{\tau_p}{J_x}$ . Se usará un PID para regular el balanceo como:

$$\tau_p = k_{p_\phi}(\phi^c - \phi) - k_{d_\phi}\dot{\phi} + k_{i_\phi} \int_0^t (\phi^c - \phi)d\tau \quad (\text{B.2})$$

El diagrama de control se muestra en la figura B.4.

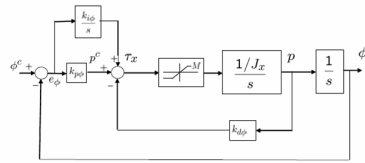


Figura B.4: Estructura de control para el balanceo

### B.3.3. Control del cabeceo

Se deriva de la misma manera que el de balanceo. Se utiliza un PID para regularlo.

### B.3.4. Control de posición para seguimiento

La dinámica lateral es dada por  $\ddot{p}_y = u_y$ , donde  $p_y$  es la posición lateral relativa que tenderá a ser cero por medio de una estrategia PID.

$$u_y = -k_{p_y}p_y - k_{d_y}\dot{p}_y + k_{i_y} \int_0^t p_y d\tau \quad (\text{B.3})$$

La dinámica longitudinal es dada por  $\ddot{p}_x = u_y$ , donde  $p_x$  es la posición lateral relativa que también será llevada a cero utilizando un PID.

$$u_x = -k_{p_x}p_x - k_{d_x}v + k_{i_x} \int_0^t p_x d\tau \quad (\text{B.4})$$

La referencia  $(x, y)$  como una posición en dos dimensiones, proporciona una lista de puntos interpolados con restricciones dinámicas entre la posición del UAV y los puntos de seguimiento o observación. sea  $uavx$  y  $uavy$  las posiciones absolutas del UAV en el entorno, el error es calculado como:

$$p_x = -x + uavx$$

$$p_y = -y + uavy$$

### B.3.5. Control de rumbo

La dinámica de la dirección es dada por  $\ddot{\psi} = \frac{\tau_p}{J_x}$ . Definiendo  $\psi^d$  como el cabeceo inercial del objetivo, si existe, y definiendo  $\tilde{\psi} \doteq \psi - \psi^d$  como dirección relativa. La cámara directamente mide  $\tilde{\psi}$ . Asumiendo que  $\psi^d$  es constante, tenemos  $\ddot{\tilde{\psi}} = \frac{\tau_p}{J_x}$ . Regulamos la dirección relativa con un PID

---

# Índice de figuras

---

1.1. Identificación y seguimiento subproblemas . . . . .	6
2.1. Trayectoria de máxima acumulación de probabilidad . . . . .	13
2.2. Trayectoria de contención . . . . .	14
2.3. Fronteras combinadas . . . . .	15
2.4. kupdate . . . . .	17
2.5. Análisis del tiempo para la contención de la frontera . . . . .	18
2.6. Algoritmo de contención . . . . .	19
2.7. Simulación de Montecarlo para contención . . . . .	22
2.8. Comportamiento de Contención . . . . .	26
2.9. Comportamiento de máxima probabilidad . . . . .	26
2.10. Comportamiento híbrido . . . . .	27
2.11. Simulación de montecarlo para caso híbrido . . . . .	28
2.12. Matriz de transición para carreteras . . . . .	29
2.13. Simulación de montecarlo para carreteras bifurcadas . . . . .	29
2.14. Simulación de montecarlo para enjambres con diferentes comportamien- tos . . . . .	30
3.1. Esquema Identificación . . . . .	31
3.2. Estados del Generador de trayectorias . . . . .	32
3.3. Función de la cámara . . . . .	34
3.4. Etapas de filtrado . . . . .	34
3.5. Diagrama del seguimiento en sentido horario . . . . .	35
3.6. Discretización del espacio para el seguimiento . . . . .	36
3.7. Identificación del contorno . . . . .	36
3.8. Extracción del contorno . . . . .	37
3.9. Detección mediante puntos de interés . . . . .	37
3.10. Reconstrucción de la imagen . . . . .	38
3.11. Simulación seguimiento de contorno . . . . .	39

3.12. Imagen reconstruida . . . . .	39
4.1. MILP sensor radial . . . . .	43
4.2. MILP sensor rectangular . . . . .	44
4.3. MILP optimización con dos UAVs . . . . .	45
4.4. MILP con dos UAVS . . . . .	45
4.5. SST modelo entradasalida . . . . .	46
4.6. SST caso 1 . . . . .	49
4.7. SST caso 2 . . . . .	50
4.8. SST caso 3 . . . . .	51
4.9. Simulación de un sensor que varía la altura . . . . .	52
4.10. Simulación de un sensor que varía la altura . . . . .	53
4.11. Múltiples agentes, contorno de dinámica lenta . . . . .	55
4.12. Múltiples agentes, resultados con dinámica lenta . . . . .	56
4.13. Múltiples agentes, contorno de dinámica rápida . . . . .	57
4.14. Múltiples agentes, resultados con dinámica rápida . . . . .	58
4.15. Comunicación local . . . . .	58
B.1. Arquitectura del sistema . . . . .	67
B.2. Diagrama de bloques del filtrado . . . . .	68
B.3. Simulación de la arquitectura . . . . .	70
B.4. Estructura de control para el balanceo . . . . .	71